Master Erasmus Mundus in
Color in Informatics and MTDMT Technology (CIMET)

Whiteboard content extraction and enhancement for videoconferencing systems

Master Thesis Report

Presented by

Carlos Andrés Arango Duque

and defended at

Gjøvik University College

Academic Supervisor(s): Jon Yngve Hardeberg
Jan Tore Korneliusen

Jury Committee:

Whiteboard content extraction and enhancement for videoconferencing systems

Carlos Andrés Arango Duque

2015/07/12

Abstract

Whiteboards have become essential for teaching, presentations and conferences since they are very flexible tools for spontaneous knowledge sharing. It's content can be captured by videoconferencing systems in order to create more engaging presentations to remote users. However depending on the lighting configuration in the scene and the position of the camera with respect to the whiteboard some shading effects and highlights may appear decreasing the contrast and legibility of the captured images. We propose a system that records and extracts the whiteboard content and enhances its appearance and legibility. Our system acquires a sequence of images from a high resolution fixed view camera and extracts the whiteboard content. Firstly we estimate the whiteboard background model using a robust surface fitting technique. We use this model to locally identify areas with written content and also to detect and remove occlusions. Secondly we estimate the illumination axis that goes through the whiteboard color distribution in the RGB color space. Finally we implement two different enhancement methods: one that balances the image colors rotating the illumination axis and another focused on enhancing the image colors using the estimated color of the illuminant. We perform a psycho-visual experiment on a dataset of images enhanced with our methods combined with other state of the art whiteboard image enhancement algorithms. The experimental results shows that our fist enhancement method provide statically more visually appealing images while our second method provide statically more legible images.

Preface

When I started this project I never imagine that it would become one of the most challenging, laborious and complex endeavours I have ever faced. This paper is the report of this long process. It might report the state of the art, theoretical knowledge, implementation details and experimental results but it cannot express the long days spent in the lab and in the library, the joy of an accomplished step, the hope for good results, the frustration after failed attempts and the immense satisfaction upon completion.

First of all I want to thank my supervisor Jon Yngve Hardeberg for his constant guidance, patience and encouragement. I want also to thank Jan Tore Korneliussen and Anders Eikenes for providing practical feedback, proofreading and dedicating part of their time and experience for the development of this project. Also I want to thank the team of Kubicam for giving me the opportunity to work with them and for providing me with their tools and knowledge. I want also acknowledge the contributions of professor Ivar Farup for helping me formulate the background and color models and professor Marius Pederson for helping me design the psychovisual experiment. I would also would like to thank Catalina Arango and Mimozë Lladrovci for they constant support and encouragement during this whole journey.

Finalmente quisiera darle las gracias a mi familia quien siempre me dan los animos y fuerzas para seguir siempre adelante y especialmente a mis padres Luis Carlos Arango y Mariela Duque y a mis tías Marta Isabel y Gloria Duque ya que sin ellos yo no estaría aquí, realizando mis sueños. Su dedicación, fortaleza, pasión y entrega han sido siempre una inspiración para mi y espero que este trabajo y otros proyectos en el futuro sean siempre motivo de orgullo para ellos.

Contents

List of Figures

List of Tables

1 Introduction

The biggest challenge in long distance communication is to transmit some information in an appropriate and understandable manner to a group of people. No matter which medium is employed, the main objective is still to be able to communicate effectively and efficiently. However videoconferencing is arguably the only medium that can appropriately convey natural human communication in real time. Not only an audience can hear what the speaker is saying but also can see the body language and feel the climate in the room and even respond with immediate feedback to the speaker [1].

Thanks to the great development in telecommunications in the last couple of decades [2] videoconferencing systems are faster and more affordable becoming a great alternative for exchanging information in education, business and media. Furthermore the speaker might want to use some visual aids such as slides or video clips to communicate his ideas in a clearer way. However they cannot be modified once they are being presented to the audience. Another aid speakers can use is a whiteboard.

Whiteboards have become essential tools for teaching and sharing information in schools, universities and companies. Since they can be quickly erased and reused they are great for spontaneous work. They can be used in many situations from explaining a math problem, displaying a process, brainstorming, and interactive exercises among others. Therefore a speaker can use a whiteboard in order to make more engaging conferences.

However there are some challenges that arise when capturing the whiteboard area with a camera. Since whiteboards are glossy smooth surfaces the whiteboard appearance is affected by the room lighting, camera position, occlusions, camera focus, etc. Also cameras with low resolution might not be able to capture fine written details resulting in blurred content. Furthermore depending on the video compression technique there is a risk of not only loosing some content details but also it can introduce some visual artifacts. Consequently the audience might have problems reading the whiteboard content.

The aforementioned challenges can be solved with a careful analysis and modelling of the extracted content. By modelling the spatial and color behaviour of the captured pixels we can formulate solutions that would give us more visually appealing images. We propose a system that extracts and enhances the contrast and color pen strokes of a whiteboard while maintaining the legibility of its content.

1.1 Context and Motivation

The proposed master thesis is a collaborative work with the start-up company Kubicam. They are a team of experienced professionals with a record of bringing high-end video communications products to market. They are currently developing an innovative videoconferencing system. Instead of designing an expensive videoconferencing system with movable cameras like some of the current commercial options in the market they are aspiring to create a more affordable sys-

Figure 1: Kubicam Logo [3]

tem aimed for universities, medium-sized and small companies among others. Their system is based on a high-resolution static camera that will focus, capture and transmit only the content of an area of interest (The location where the speaker is, an specific object or area that the speaker want to focus on, etc.). Our proposed system is framed under this line of research and we are aiming to develop an application that can be integrated with the overall videoconferencing system of Kubicam.

1.2 Objectives

The main objectives that we aim to achieve during this work are:

- Review the state of the art in whiteboard content extraction and enhancement
- Analyze and model the spatial and color behaviour of a whiteboard and its content.
- Classify whiteboard areas as written content, background and occlusion.
- Enhance the appearance and legibility of the whiteboard image content.
- Test, analyse and compare our results with state of the art algorithms for whiteboard enhancement.
- Present results, new challenges and establish the lines for future work.

2 State of the Art

2.1 Whiteboard Content Extraction

A careful review of the state of the art revealed a great interest of the scientific community in researching ways to extract and process the content of whiteboards. However we observed that many authors had different goals in mind consequently their work was focused into solving different tasks.

For instance some authors focused only in detecting and segmenting the pen strokes for handwritten recognition. Lu and Kowalkiewicz [4] proposed a method that detected whiteboard pen strokes using an edge tracing algorithm and identified which ones were handwritten letters. Plötz et al. [5] designed a system that detected text regions, performs a globally optimized threshold comparison to increase the pen strokes contrast and finally recognized the handwritten text. Wienecke et al. [6] proposed a method where they divide the whiteboard image into blocks, detect the areas with pen strokes, discriminate between foreground and background using a modified version of Niblack binarization [7], extract features from the text and proceed to recognize it. Vajda et al. [8] go a step further by proposing a system that recognizes mind maps. It does so by classifying pen strokes as text, lines, circles and arrows and associate groups of letters as words by density based clustering.

Other authors focus in creating applications that extract a series of content-rich key frames in a classroom or meeting room without interrupting the presentation. In these approaches the two main goals are to remove the speaker from the image and detect when the speaker has written new content or erased previous one. Fakih and Serrao [9] propose a simple method where the first frame is used as reference and as whiteboard background model and then any obtained content is used to update the reference frame. The work from Rao et al. [10] tracks the speaker by means of median background subtraction and replace this area with information from previous frames. Dickson et al. [11] create an average image by dividing the input image into small blocks and the 25% brightest pixels are determined and used to create the average color value for the block. This average image is used to refine the whiteboard input image and the speaker is tracked by the difference of two consecutive refined images. Xu [12] propose an interesting approach using a static and a Pan Tilt-Zoom (PTZ) cameras. The static camera is used to detect changes in the whiteboard and the position of the speaker while the PTZ camera scans and extract the newly updated whiteboard regions.

2.2 Whiteboard Content Enhancement

The applications shown in the previous section focused on the whiteboard content extraction step for text recognition or key frame detection. However there are some instances where the user would like to share the content with other people. In these cases we need to improve the quality and legibility of the whiteboard content. In this section we will review some of the methods used

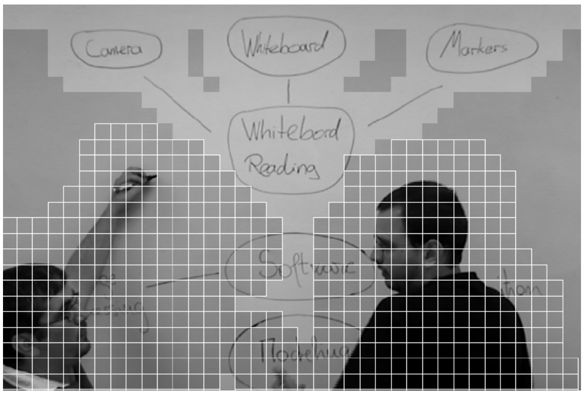


Figure 2: Vajda et al. method for recognizing mind maps. The image is segmented into text, background and occlusions [8]

for these purposes.

Sakshuwong and Tsai [13] proposed an iPhone application that globally thresholds the whiteboard image in real time, and segments and enhances the image on demand when the user decides to save an image. The first step is done by comparing each pixel to a weighted average of its neighbours and then a global threshold is computed using Otsu's method [14]. The second step consists in looking up the color information of the selected pixels and clustering into color groups. It is implied that they transform the color values into the rg chromaticity space [15] and ignore low saturated colors in order for the k-means clustering algorithm to be effective. After the clustering is performed each pixel is compared to its neighbours and it is reclassified into the majority color.

He et al. [16] proposed a method that directly addresses the highlight specular reflection problem in whiteboards. First the whiteboard image pixels are separated into specular or diffuse based on the dichromatic reflection model [17] and the Specular-to-Diffuse mechanism [18]. Second the specular pixels are enhanced by decreasing their intensity to their diffuse pixel equivalent and the diffuse pixels saturation is increased. Third the stroke-strength of each pixel is estimated from the original image to indicate the degree of a pixel to be a pen-stroke. Finally the pen-strokes intensities are reduced according to the estimated stroke-strength and maximum chromaticity.

Gormish et al. [19] proposed an approach of segmenting the whiteboard image between background and pen-strokes, where they remove the background and enhance the pen-strokes. The background is modelled by applying a local median filter to the original image. Then the background image is subtracted from the original image. The pen strokes edges are detected using a Canny edge detector [20] and then they are morphologically dilated to avoid creating a pen-stroke mask with double edges[1]. Finally the segmented edges are enhanced by adjusting their saturation using a sigmoid function or S-shaped fitting curve (C and k are constants set by

[1] The problem of segmenting thin lines using Canny edge detector is that the algorithm will draw a line wrapped around the line giving the illusion of double edges

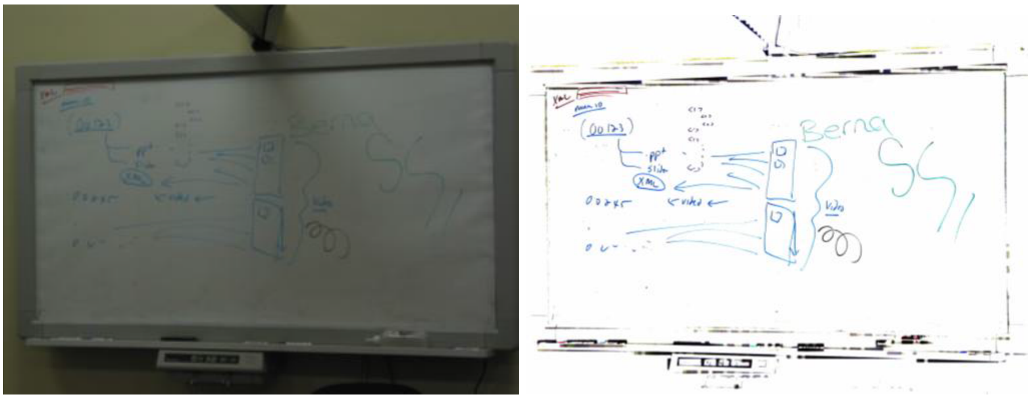


Figure 3: (left) Whiteboard image captured by a digital camera (right) enhanced with the Gormish et al. method [19]

the authors):

$$R' = \frac{1}{1+e^{C(k-R)}} \qquad G' = \frac{1}{1+e^{C(k-G)}} \qquad B' = \frac{1}{1+e^{C(k-B)}} \tag{2.1}$$

Perhaps the most documented and complete work on whiteboard image capture and enhancement is the one from Zhengyou Zhang and Li-wei He. Although their system has changed over the years the main workflow is as follows [21]: First they automatically detect the borders of the whiteboard in the image. This is done by detecting the four strong edges that define the whiteboard quadrangle. Secondly they estimate the aspect ratio of the whiteboard from the detected quadrangle, calculate the homography matrix from the original image to a rectangular image with the estimated aspect ratio and the whiteboard image is rectified accordingly. Thirdly they estimate the background color under the assumption that if the whiteboard is divided into small cells, the whiteboard cells have the brightest luminance over time and have small variance [22]. This technique was later updated. Assuming that color varies smoothly across the whiteboard a plane can be fitted in the luminance Y or RGB color space by minimizing the median of squared errors [23]. The whiteboard background is scaled using the estimated background making it uniformly white. Finally the pen strokes saturations is adjusted using an S-Shaped fitting curve (p is a contant set by the authors):

$$R_{out} = 0.5 - 0.5\cos\left(R^{p} * \pi\right) \qquad G_{out} = 0.5 - 0.5\cos\left(G^{p} * \pi\right) \qquad B_{out} = 0.5 - 0.5\cos\left(B^{p} * \pi\right) \tag{2.2}$$

2.3 Commercial Applications

There are some commercial applications for capturing and enhancing whiteboard content on the internet. For many years Pixid Whiteboard Photo [24] [25] was the standard software although it appears it has been discontinued.There is also a software called ClearBoard [26] with a tag price of 99.95 US$. Snapclean.me [27] and (un)whiteboard [28] are web apps that can do the whiteboard conversion online. Furthermore there is a variety of cellphone apps available that can do the same thing for a very small price [29]. There also some tutorials for enhancing whiteboard

Figure 4: Microsoft newest interactive board: the Surface Hub [41]

pictures using Photoshop [30] [31]. The problem with these approaches is that they only work for static images previously taken by the user.

FXPAL has also developed a system called ReBoard [32] that not only captures the whiteboard data but also retrieves a range of metadata including date, presence of collaborators during content creation, relative amount of flux during content creation, spatial location of content on the board among other information [33].

2.4 Interactive Whiteboards

Another option comes from interactive whiteboards also known as smart boards. These tools allow computer images to be displayed onto a board. The speaker can then manipulate the elements displayed and make annotations directly onto the board [34]. There are two types of smart boards:

- **Projector boards:**These ones project the images onto a whiteboard and also uses a CMOS camera to detect the position of an IR light or a motion-sensing pen which is pressed against the whiteboard [35]. There are several companies offering these solutions such as the SMART board [36], the Promethean Active Board Touch [37] and the Luidia Inc. eBeam [38] to name a few.
- **Interactive Displays:**They are large interaction displays (they may be touchscreens) that are connected to a computer and mounted to a wall or floor stand. The user can interact with the screen either using their fingers, pen or other IR light pointer [39] [40]. Microsoft is aiming to lead the interactive whiteboard market with their new SurfaceHub, an 84-inch 4K interactive HDTV [41] [42].

3 Whiteboard Analysis

Before we propose any solution we must give a closer look at the problem at hand. The fist step is to analyse different aspects of a whiteboard in order to understand what challenges and obstacles that we must overcome are. We must then ask a series of questions:

- How do whiteboards and dry-erase markers work?
- What are the challenges of capturing a whiteboard image?
- Can we model the spatial behaviour of a whiteboard?
    1. Can we describe smooth changes of luminance across the whiteboard?
    2. What about specular reflection?
- Can we model the color behaviour of a whiteboard?
    1. Do whiteboard pixels show a pattern in a color space?
    2. Can we estimate the pen-stroke colors following these patterns?

In this section we will try to answer these questions and establish a mathematical base for the solutions that will be presented in the following chapter.

3.1 Understanding the problem

3.1.1 Whiteboards

Whiteboards are white glossy surfaces used for non-permanent markings. They come in different materials such as laminated chipboards, melanine, painted aluminium and porcelain steel [43]. Whiteboards have become an essential tool for schools, universities and organizations. Their biggest advantage is that they can be quickly erased and reused making them a very flexible tool for spontaneous knowledge sharing. They can be used for explaining a math problem, displaying a process, brainstorming, and interactive exercises among others.

Furthermore they are cleaner than traditional blackboards since they use dry erasable markers instead of chalks that produce dust which may affect people with dust allergies [44]. Also they can be used as a projecting medium when using a video projector and we can comment, underline and highlight important details using a marker [45].

3.1.2 Dry erase markers

Also called non-permanent marker, are markers designed to be wiped easily off non-porous surfaces, specifically whiteboards. The ink used in dry-erase markers is very similar to that used in permanent markers. It is comprised of color pigments, a chemical solvent (alcohol) and a polymer (“release agent”). However, while permanent markers use acrylic polymers in order to make the pigments sticks to the surface, dry erase markers use an oily silicone polymer that prevents the colored pigments from coming in direct contact with the surface. Furthermore, the use of alcohol as solvent makes the ink dry quickly, becoming attached to the surface but not absorbed [46].

Dry erase markers are meant to be used in non-porous surfaces like glass and metal (the ink becomes permanent in porous surfaces like paper or skin). However they are mostly suited for whiteboards since they have a slightly charged surface that helps the pigments to adhere to the surface [47]. Dry erase marker standard colors are black, red, green and blue. However companies nowadays provide a rainbow of different colors for dry erase markers including orange, purple, brown, etc.

3.1.3 Challenges

There are some disadvantages of using whiteboards. The first one is that if you write in a whiteboard using the wrong type of marker it might be difficult to erase and that will make the whiteboard dirty. The second one is that it releases some chemical vapors and after prolonged exposure small children and people with chemical sensitivities may experience throat and eye irritation [48]. The third one is that it is more expensive than traditional chalks and unlike chalk visibly diminishing size we cannot see when the marker is running out of ink.

Finally and most importantly is that since we are dealing with a glossy surface, its appearance is heavily affected by the environmental lighting. Depending on the location of the light sources and the observer with respect of the whiteboard, we might encounter differences in shading along the whiteboard and glare or specular highlights. These two undesirable effects may decrease the contrast and legibility in whiteboards.

The latter effect is minimized in human observers thanks to the ability of our visual system to maintain object color appearance across variations(color constancy) [49]. However it can become problematic when we need to capture its content with a camera. Although it is possible to minimize this problem with a professional lighting set-up [50], it requires the use of professional lighting equipment for photography and might not be suitable for most applications.

3.2 Whiteboard Spatial Modelling

3.2.1 Problem Formulation

The first step of addressing the aforementioned challenges is to model the shading effect which is caused by inhomogeneous illumination conditions. In the literature the connection between an assumed shading-free image $U(x, y)$ and the image $I(x, y)$ which is acquired by a real camera has been formulated as a linear model [51]:

$$I(x, y) = U(x, y)S_M(x, y) + S_A(x, y) \tag{3.1}$$

The components $S_M(x, y)$ and $S_A(x, y)$ are multiplicative and additive shading components respectively and they are spatially variable. If the shading components are known, shading correction is achieved by inverting the linear model (Equation 3.1):

$$U(x, y) = \frac{I(x, y) - S_A(x, y)}{S_M(x, y)} \tag{3.2}$$

Normally the shading correction is simplified by assuming that only one of the components is involved in the image shading. In such case the shading correction is calculated either by:

$$U(x, y) = I(x, y) - S_A(x, y) + C \tag{3.3}$$

or

$$U(x,y) = \frac{I(x,y)}{S_M(x,y)} \tag{3.4}$$

Where the normalization constant C can be computed as the mean color of the shading estimate (However this step is not necessary since we are mostly interested in the shading model).

3.2.2 Shading Correction Techniques

- **Linear and homomorphic filtering:** It assumes that shading is located in the low frequency domain below the image content of interest and only cares for the additive shading component $S_A(x,y)$. It can be estimated by applying a low-pass filter [52], a Gaussian filter [51] or a local median filter [19] to the acquired image.
- **Morphological Filtering:** it is based on the assumption that objects of interest are smaller than the background variations and that the background is either darker or lighter than the objects all around the image. Both $S_A(x,y)$ and $S_M(x,y)$ can be estimated by applying greyscale morphologic operators to the acquired image.
- **Fitting a shading model:** The shading model can be estimated by selecting a number of points in the background. The intensity variation over the image background may be obtained by least-squares fitting the selected points into a plane, uqadratic surface, etc. This technique is based on the assumption that all the selected points are representative values (not outliers) and are well distributed across the image [53].
- **Entropy Minimization:** it is assumed that the image formation model (Equation 3.1) increases the level of entropy in the image. Shading correction is performed by modelling a fitting a second degree polynomial $S_A(x,y)$ or $S_M(x,y)$ and varying its parameters until the entropy H of the image $U(x,y)$ reaches a minimum:

  $$U_o(x,y) = \min(H(I(x,y))) \tag{3.5}$$

  where $U_o(x,y)$ is the optimal corrected image

3.2.3 Plane Fitting

Let's focus on how can we apply the shading model to whiteboards. Since whiteboards are physically built to be uniformly white we can simplify the shading model equation considering only its multiplicative component S_M (Equation 3.4). Let's take a closer look to the method proposed by Zhang et al. [21].The first step is to divide the whiteboard region into rectangular cells in order to low the computational cost. This is based on the assumptions that a significant portion of the pixels in each cell belongs to the background(These pixels have the brightest luminance) and that the luminance in each cell have a small variance (the luminance in the cell is almost uniform). Therefore the background value for each cell is calculated as the value of its brightest pixels. This works even if there is a stroke throughout the cell. However this method will produce some holes in the modelled whiteboard image when a cell is blocked by a foreground object or colored filled shapes. They called these holes outliers. The next step is to detect these outliers by fitting a plane for the whole whiteboard area in the luminance plane.

A plane can be represented by $ax + by + c - z = 0$. The whiteboard cells can be seen as a set of 3D points $\{(x_i, y_i, z_i) | i = 1, \ldots, n\}$ where x_i and y_i are the cell coordinates and z_i is the

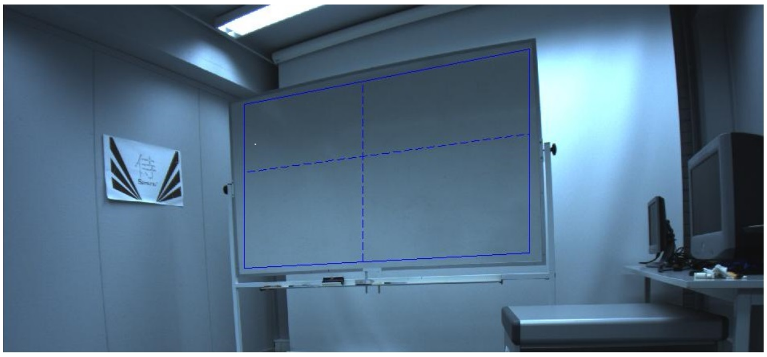


Figure 5: Empty whiteboard. The blue continuous lines specify the sample area. The blue dashed lines represent 2 samples taken for analysis

luminance. The plane parameters represented as $\mathbf{p} = [a, b, c]^T$ can be estimated by minimizing the following function:

$$F = \min \sum_i f_i^2 \qquad \text{where} \qquad f_i = ax_i + by_i + c - z_i \tag{3.6}$$

The least squares closed solution is given by:

$$\mathbf{p} = (A^T A)^{-1} A^T z \qquad \text{where} \qquad A = \begin{bmatrix} x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots \\ x_n & y_n & 1 \end{bmatrix} \qquad \text{and} \qquad z = \begin{bmatrix} z_1 \\ \vdots \\ z_n \end{bmatrix} \tag{3.7}$$

Once the plane parameters are determined the color of the cells are calculated by:

$$\hat{z}_i = ax_i + by_i + c \tag{3.8}$$

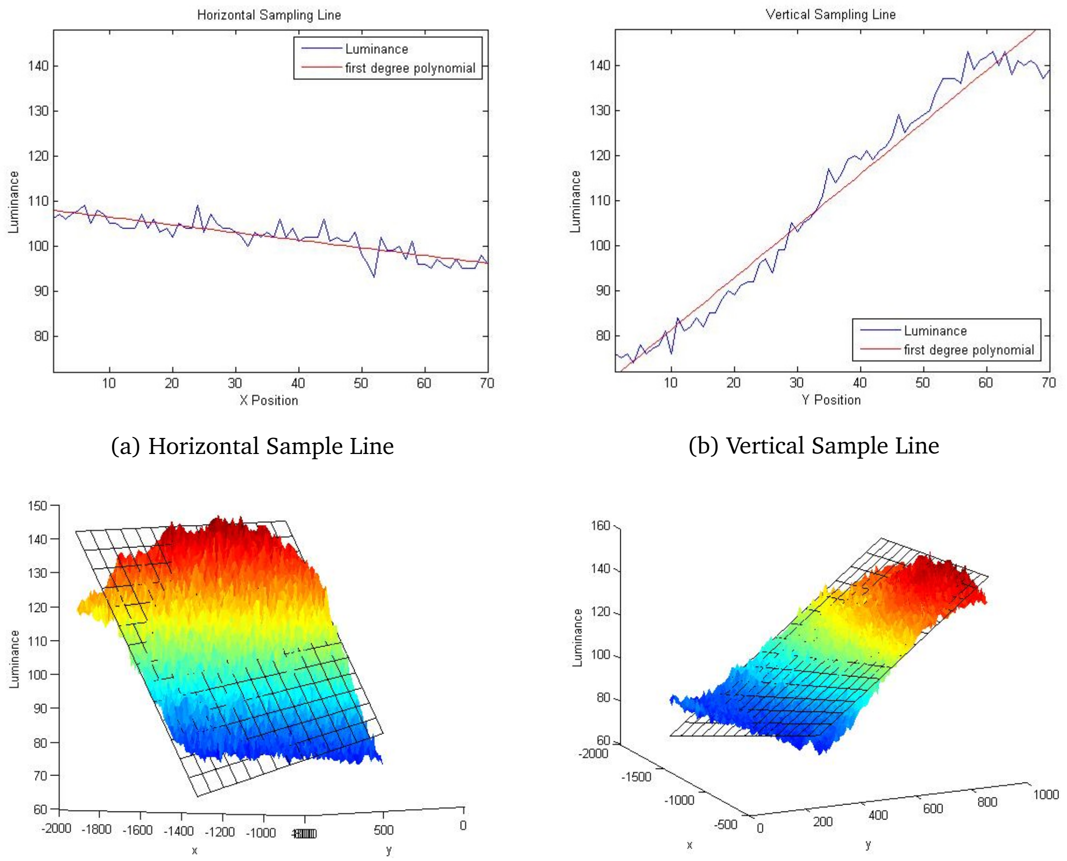
3.2.4 Surface Fitting

The method in Section 3.2.3 is based on the assumption that the luminance or color components of the whiteboard varies similarly to a plane. However, a closer inspection at whiteboards show us that their behaviour is more complex. Let's take for example Fig.5. We extract the luminance values inside the blue quadrilateral. First let's take a couple of samples: one indicated by the vertical dashed line and another indicated by the horizontal dashed line. As we can see in Fig.6a the luminance behaviour can be properly modelled using a line equation. However in Fig.6b we can see the luminance has a more curved behaviour.

This discrepancy is even more noticeable when we try to compare the whiteboard luminance surface with the estimated plane in 3D graphics (Fig.6c and Fig.6d). We can notice that a plane is ill-suited to model the whiteboard luminance especially in the corners of the whiteboard.

Let's take a different approach. If we inspect the Fig.6a and Fig.6b again we can consider using a polynomial of 2nd or 3rd degree to model the luminance. As we can see in Fig.7. a 3rd degree polynomial equation fits better for both scenarios.

Since we want to fit a 3rd degree polynomial surface to estimate the whiteboard luminance we need to modify the method described in Section 3.2.3. Now we need to solve Equation 3.6



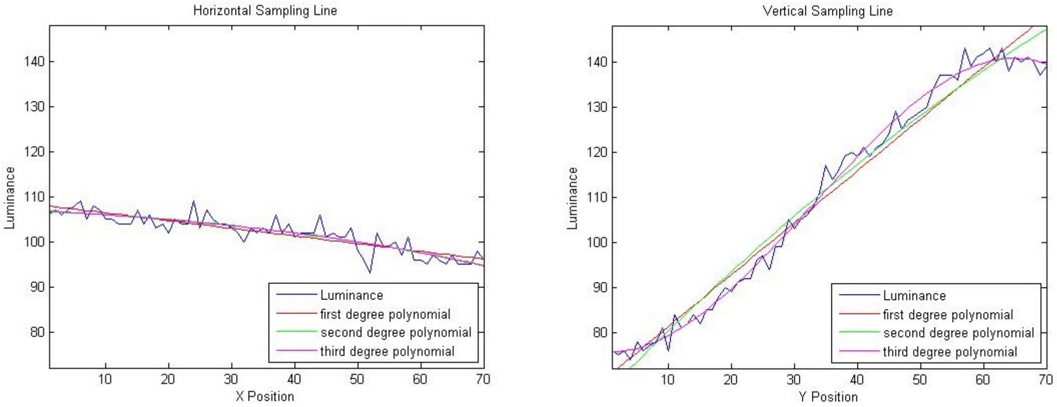
(a) Horizontal Sample Line

(b) Vertical Sample Line

(c)

(d)

Figure 6: Plane Fitting the whiteboard luminance. (Up) 2D representation. The blue line represents the sampled luminance, The red line the estimated luminance. (Down) 3D representation. The coloured surface represents the sample luminance and the transparent plane represents the estimated luminance.



(a) Horizontal Sample Line

(b) Vertical Sample Line

Figure 7: The blue line represents the sampled luminance, the red line the first degree polynomial, the green line second degree polynomial and the purple line the third degree polynomial

where:

$$f_i = ay_i^3 + bx_i^3 + cx_i^2y_i^2 + dx_i^2y_i + ex_iy_i^2 + fy_i^2 + gx_i^2 + hx_iy_i + ix_i + jy_i + k - z_i \tag{3.9}$$

As we can see in Fig.8 the proposed surface fits better the whiteboard luminance even near the corners. However the least-squares technique is very sensitive to noise. As we mention earlier the whiteboard might contain outliers (glare, occlusion, drawings). For example Fig.9a shows a captured whiteboard image with two strong specular reflection areas. If we take the horizontal dashed line as sample we can see that we fail to correctly fit the luminance(Fig.9b). In order to detect and reject these outliers we use a robust technique to fit the whiteboard image called Least Median Squares (LMedS) [23]. The idea is to minimize the median rather than the sum of squared errors:

$$F = \min \operatorname*{med}_{i=1,\ldots,n} f_i^2 \tag{3.10}$$

However this cannot be solved with a closed solution. Rousseeuw [54] proposed a Monte Carlo type technique that finds a solution by random subsampling. The procedure is as follows [55]:

- Choose N random subsamples of $\mathbf{p}$ different data points (In our case $\mathbf{p} = 10$ since we are trying to fit a 3rd degree polynomial surface)
- For each subsample $\mathbf{p}_j$ we calculate the estimate of the plane using the Least Squares closed solution (Equation 3.7)
- For each subsample $\mathbf{p}_j$ we determine the median of the squared residuals:

$$M_j = \operatorname*{med}_{i=1,\ldots,n} f_i^2(\mathbf{p}_j, z_j) \tag{3.11}$$

- We retain the estimate $\mathbf{p}_j$ for which M_j is minimal.

However how do we determine the number of random samples (N)? Rousseeuw and Leroy [56] proposed that assuming that the whole set of points contains a fraction of ε of outliers, the probability(P)that at least one of the N subsamples is good is given by:

$$P = 1 - 1(1 - (1 - \varepsilon)^p)^N \tag{3.12}$$

We can determine that N is :

$$N = \frac{\log(1 - P)}{\log(1 - (1 - \varepsilon)^p)} \tag{3.13}$$

For example if we would like to find a good set of points to model a plane where 50% of the data might be outliers with 99% of accuracy we will need at least 4714 subsamples. However [56] pointed out that LMedS efficiency is poor in presence of Gaussian Noise. To compensate for this deficiency the next step is to carry a weighted least-squares procedure. We calculate the so-called robust standard deviation:

$$\hat{\sigma} = 1.4826\sqrt{M_j} \tag{3.14}$$

where M_j is the minimal median and the constant 1.4826 is a coefficient to achieve the same efficiency as a least-squares in the presence of only Gaussian noise (when there are no outliers present). Based on $\hat{\sigma}$ we assign the weights for each point:

$$\omega_i = \begin{cases} 1 & if \quad f_i^2 \leq (2.5\hat{\sigma})^2 \\ 0 & otherwise \end{cases} \tag{3.15}$$

We fit a plane by solving the weighted least-squares problem:

$$F = \min_p \sum_i \omega_i f_i^2 \tag{3.16}$$

Finally the color of any outlier cell i is replaced using Equation 3.8. As we can see in Fig.9c and Fig.9d the new model is able to fit a surface to the whiteboard luminance while ignoring highlights.

3.3 Whiteboard Color Modeling

3.3.1 Background Color

In this section we will try to model the whiteboard background color values. Let's take for example the whiteboard in Fig.5 and analyse it's variation in the RGB color space. The color values of the whiteboard can be seen as a cloud of points inside a RGB color cube. We are dealing with a smooth white surface with a shading component so that would mean in theory that the color values should concentrate near the neutral gray axis [57]. However as we can see in Fig10 the whiteboard color distribution behaves as a contained ellipsoidal distribution with its semi-major axis oriented around an unknown axis. This axis named illumination axis ($\mathbf{e}_l$) is defined by the color spectrum of the illuminant that has been used rather than the gray axis [58].

This can be done assuming that there is only one illuminant in the scene(or at least one that covers most of the whiteboard area). We can estimate the illumination axis as the intersection of two planes, one parallel to the R-axis and one to the G-axis:

$$B = m_{rb}R + c_{rb} \qquad \text{and} \qquad B = m_{gb}G + c_{gb} \tag{3.17}$$

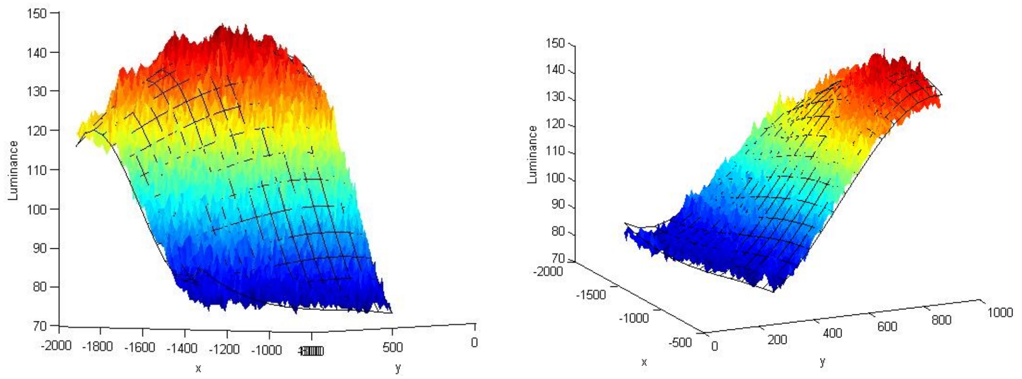
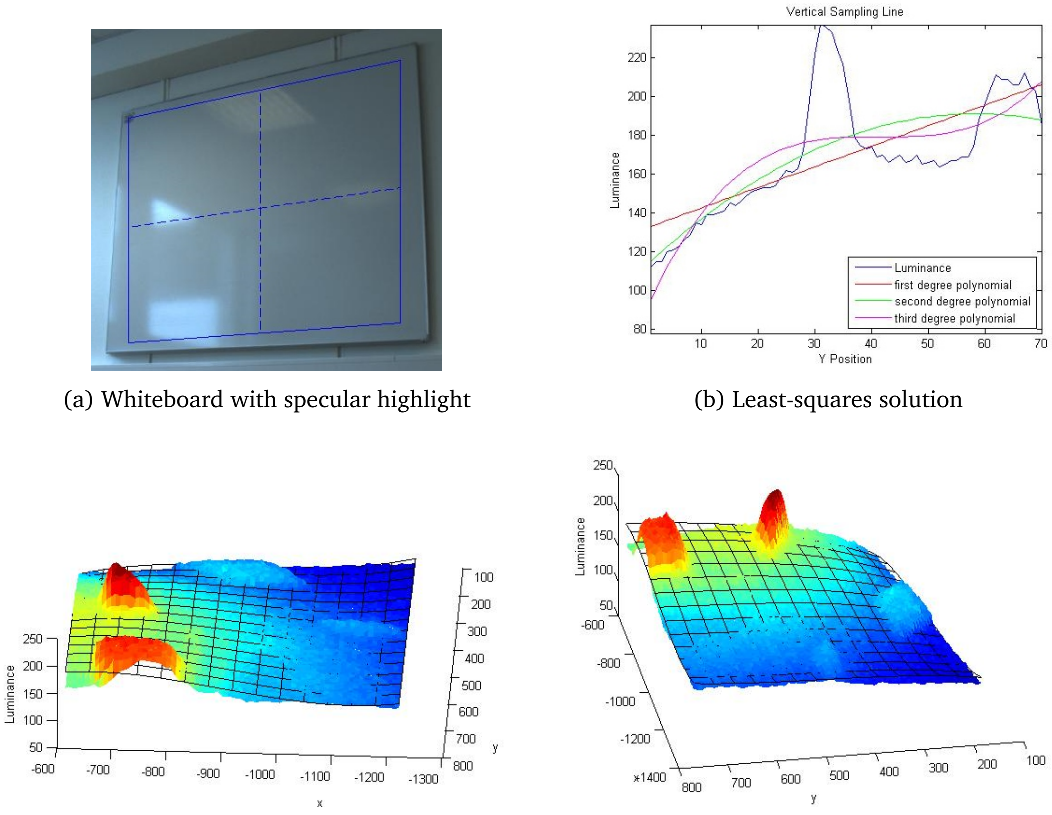


Figure 8: Surface Fitting. The coloured surface represents the sample luminance and the transparent surface represents the estimated luminance



(a) Whiteboard with specular highlight

(b) Least-squares solution

(c) LMedS solution

(d) LMedS solution

Figure 9: Fitting a surface in presence of highlights using LMedS

We can find the planes parameters (slope and intersect) that best fit our color values by representing our data as:

$$\begin{bmatrix} R_1 & 1 \\ R_2 & 1 \\ \vdots & \vdots \\ R_n & 1 \end{bmatrix} \begin{bmatrix} m_{rb} \\ c_{rb} \end{bmatrix} = \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_n \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} G_1 & 1 \\ G_2 & 1 \\ \vdots & \vdots \\ G_n & 1 \end{bmatrix} \begin{bmatrix} m_{gb} \\ c_{gb} \end{bmatrix} = \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_n \end{bmatrix} \tag{3.18}$$

The form of Equations 3.18 are $Ax = b$ and they can be solved individually with a least squares estimation (Fig.11). The intersection of those two planes and the function for the illumination axis would be:

$$R = \frac{B - c_{rb}}{m_{rb}} \quad \text{and} \quad G = \frac{B - c_{gb}}{m_{gb}} \tag{3.19}$$

As we can see in Fig.12 the illumination axis describes the whiteboard background color orientation in the color space. Furthermore we can calculate the color of the illuminant (W_e) by finding the color value for which the illuminant has single-channel saturation. This is done by simply intersecting the illumination axis with the RGB color space cube:

$$W_e = \begin{cases} W_r = \left(1, \frac{m_{rb}+c_{rb}-c_{gb}}{m_{gb}}, m_{rb}+c_{rb}\right) & \text{if} \quad max(W_r) <= 1 \\ W_g = \left(\frac{m_{gb}+c_{gb}-c_{rb}}{m_{rb}}, 1, m_{gb}+c_{gb}\right) & \text{if} \quad max(W_g) <= 1 \\ W_b = \left(\frac{1-c_{rb}}{m_{rb}}, \frac{1-c_{gb}}{m_{gb}}, 1\right) & \text{if} \quad max(W_b) <= 1 \end{cases} \tag{3.20}$$

We can use the same procedure to calculate the value of the darkest color (Bl_e) for the axis:

$$Bl_e = \begin{cases} Bl_r = \left(0, \frac{c_{rb}-c_{gb}}{m_{gb}}, c_{rb}\right) & \text{if} \quad min(Bl_r) >= 0 \\ Bl_g = \left(\frac{c_{gb}-c_{rb}}{m_{rb}}, 0, c_{gb}\right) & \text{if} \quad min(Bl_g) >= 0 \\ Bl_b = \left(\frac{c_{rb}}{m_{rb}}, \frac{c_{gb}}{m_{gb}}, 0\right) & \text{if} \quad min(Bl_b) >= 0 \end{cases} \tag{3.21}$$

We can also see Bl_e as the shift between the illumination axis and the neutral gray axis. Once we got this values we can calculate the illumination axis vector ($\mathbf{e}_i$).

$$\vec{\mathbf{e}_i} = \frac{W_e - Bl_e}{\|W_e - Bl_e\|} \tag{3.22}$$

3.3.2 Pen-stroke color

In Section 3.3.1 we were able to model an axis that describe the color distribution of the whiteboard colors but what about the dry-erase marker colors? In theory color values heavily depend on both illumination intensity and the color of the illuminant. Moreover, in presence of non-white illumination, object surfaces exhibit illumination-induced color changes [59]. So we could assume that the marker colors can be modelled around the illumination axis. Let's take the whiteboard image in Fig.13a as example. As we can see is a classical whiteboard image with handwriting made with different colors of markers (black,green,blue and red). We segment

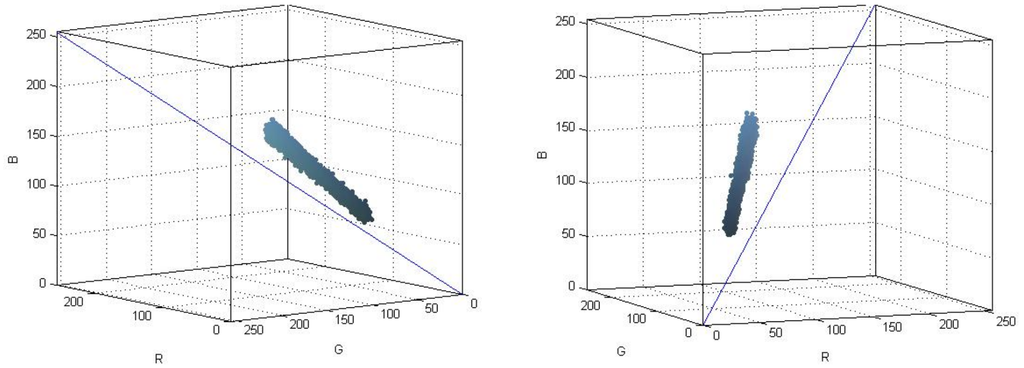


Figure 10: Whiteboard background color distribution. The blue line represents the neutral gray axis

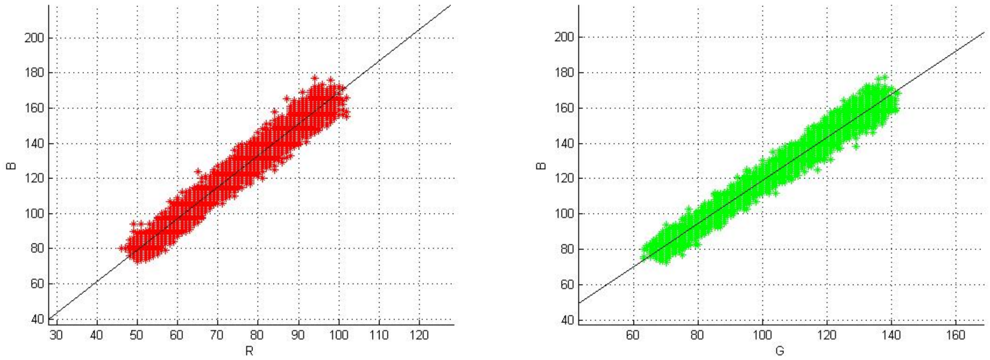


Figure 11: (Left)The black line represents the estimated plane parallel to the G-axis (Right)The estimated plane parallel to the R-axis

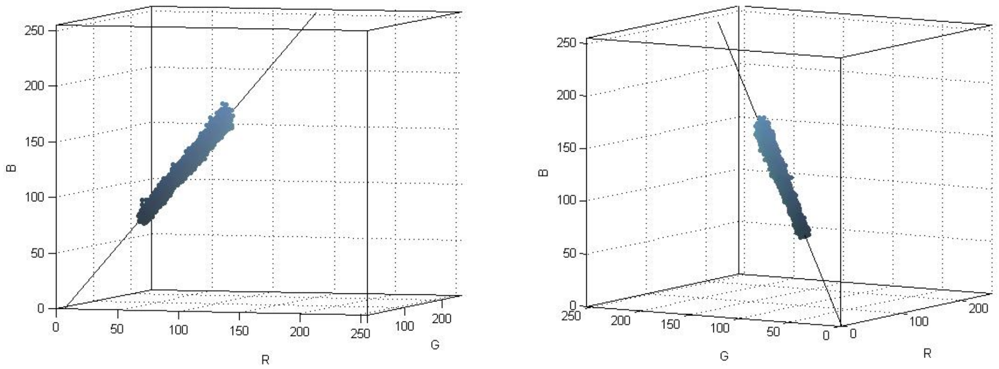


Figure 12: Whiteboard background color estimation. The black line represents the illumination axis

the pen-strokes[1] and we calculate the illumination axis using the whiteboard background. Then we take the color of the segmented pen-strokes and represent them into the RGB color space (Fig.13c and Fig.13d). As we can see the different marker colors are distributed in concentrated clouds around the illumination axis. Furthermore we can represent any color as a vector using the illumination axis. First we take a color value ($F_e = [R_e, G_e, B_e]$) and we calculate its vector with respect of the origin of the illumination axis $\overrightarrow{F_e} = Col_e - Bl_e$. Now we can calculate the orthogonal projection of $\overrightarrow{F_e}$ onto the illumination axis [60]:

$$\overrightarrow{L_e} = L_e \overrightarrow{e_i} \qquad \text{where} \qquad L_e = \overrightarrow{F_e} \bullet \overrightarrow{e_i} \tag{3.23}$$

where L_e is the scalar projection of $\overrightarrow{F_e}$ onto $\overrightarrow{e_i}$ and can be considered as the color lightness. We can also calculate the vector rejection. By definition:

$$\overrightarrow{C_e} = \overrightarrow{F_e} - \overrightarrow{L_e} \tag{3.24}$$

thus:

$$\overrightarrow{C_e} = \overrightarrow{F_e} - (\overrightarrow{F_e} \bullet \overrightarrow{e_i})\overrightarrow{e_i} \tag{3.25}$$

Since $\|C_e\|$ denotes the shortest euclidean distance from Col_e to the illumination axis we can consider it as chroma. Finally we can represent any color as the sum of its vector projection and rejection (Fig.14a):

$$\overrightarrow{F_e} = \overrightarrow{L_e} + \overrightarrow{C_e} \tag{3.26}$$

We can also calculate the Hue by transforming our colors to the rg chromaticity space. First we need to shift our colors so the illumination axis starts at the same point as the neutral gray axis (origin).

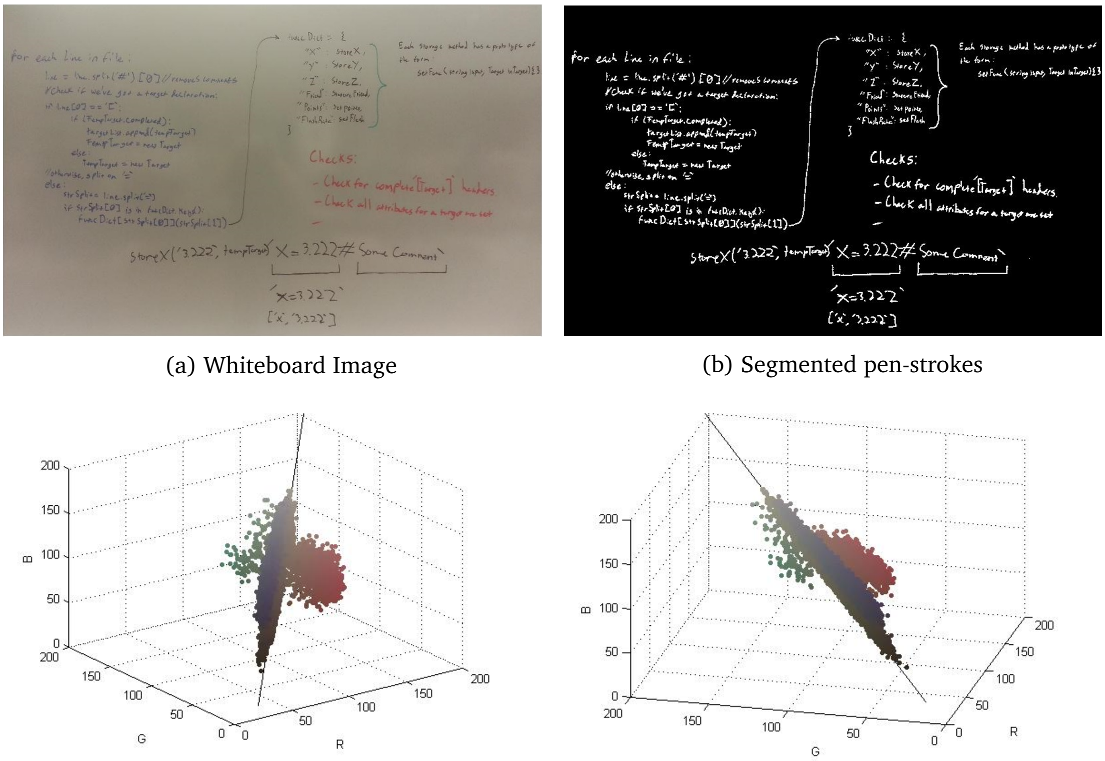
$$r_{F'} = \frac{R'_e}{R'_e + G'_e + B'_e} \qquad g_{F'} = \frac{G'_e}{R'_e + G'_e + B'_e} \qquad \text{where} \qquad \overrightarrow{F_e} = [R'_e, G'_e, B'_e] \tag{3.27}$$

The Hue value (H_e) is calculated as the angle of the chromaticity value using the shifted illumination axis white value (W_e) as the rotation axis:

$$r_{W'} = \frac{W'_r}{W'_r + W'_g + W'_b} \qquad g_{W'} = \frac{W'_g}{W'_r + W'_g + W'_b} \qquad W'_e = W_e - Bl_e \tag{3.28}$$

$$H_e = \arctan\left(\frac{g_{F'} - g_{W'}}{r_{F'} - r_{W'}}\right) \tag{3.29}$$

[1] The image used for this example is exceptionally easy to segment. However this level of refined segmentation is difficult to achieve for most whiteboard images



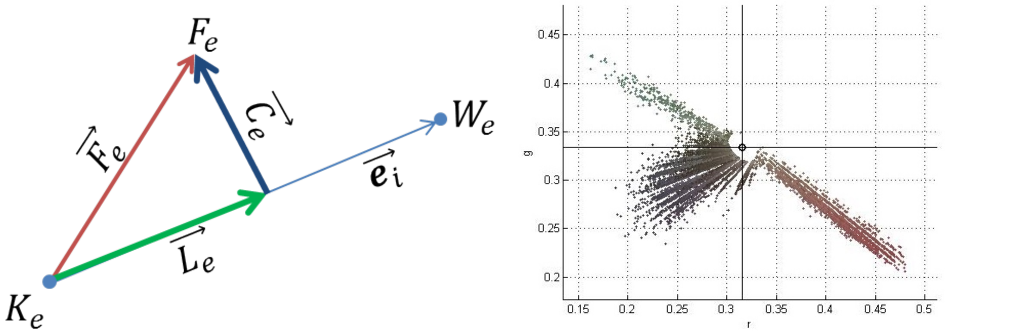
(a) Whiteboard Image

(b) Segmented pen-strokes

(c) Pen-stroke colors

(d) Pen-stroke colors

Figure 13: Pen strokes color estimation. The black line represents the illumination axis



(a) Chroma and Lightness calculation. Each color is calculated w.r.t the illumination axis

(b) Hue calculation in the rg chromaticity space. Each angle is calculated w.r.t the shifted white value

Figure 14: Pen-stroke color parameters

4 System Architecture

In this chapter we will describe in detail the materials and steps of our proposed system. The first section describes the hardware and software used for the system development. The second section describes how we extract and rectify the whiteboard area from the acquisition system. The third and fourth section describe the implementation details of the color and background modelling respectively. Finally the fifth section will describe the image background and pen-stroke enhancement process.

As suggested by our flowchart in Fig.15b the system waits for the user to select the whiteboard area before it crops and rectifies the image. Then it asks whether the system is calibrated before doing an initial color modelling to estimate the whiteboard background and the illumination axis. This step is only executed once unless the user request to recalibrate the system (In case of unsatisfactory results or if the lighting conditions have changed). The next step is to classify the image cells using the rectified image and the estimated background. This step along with the image rectification step are executed every cycle. The final step is to use the cells classified as pen-strokes, the modelled background and the input image to create an enhanced version of the input image. This last step is done upon user request.

4.1 Tools and Materials

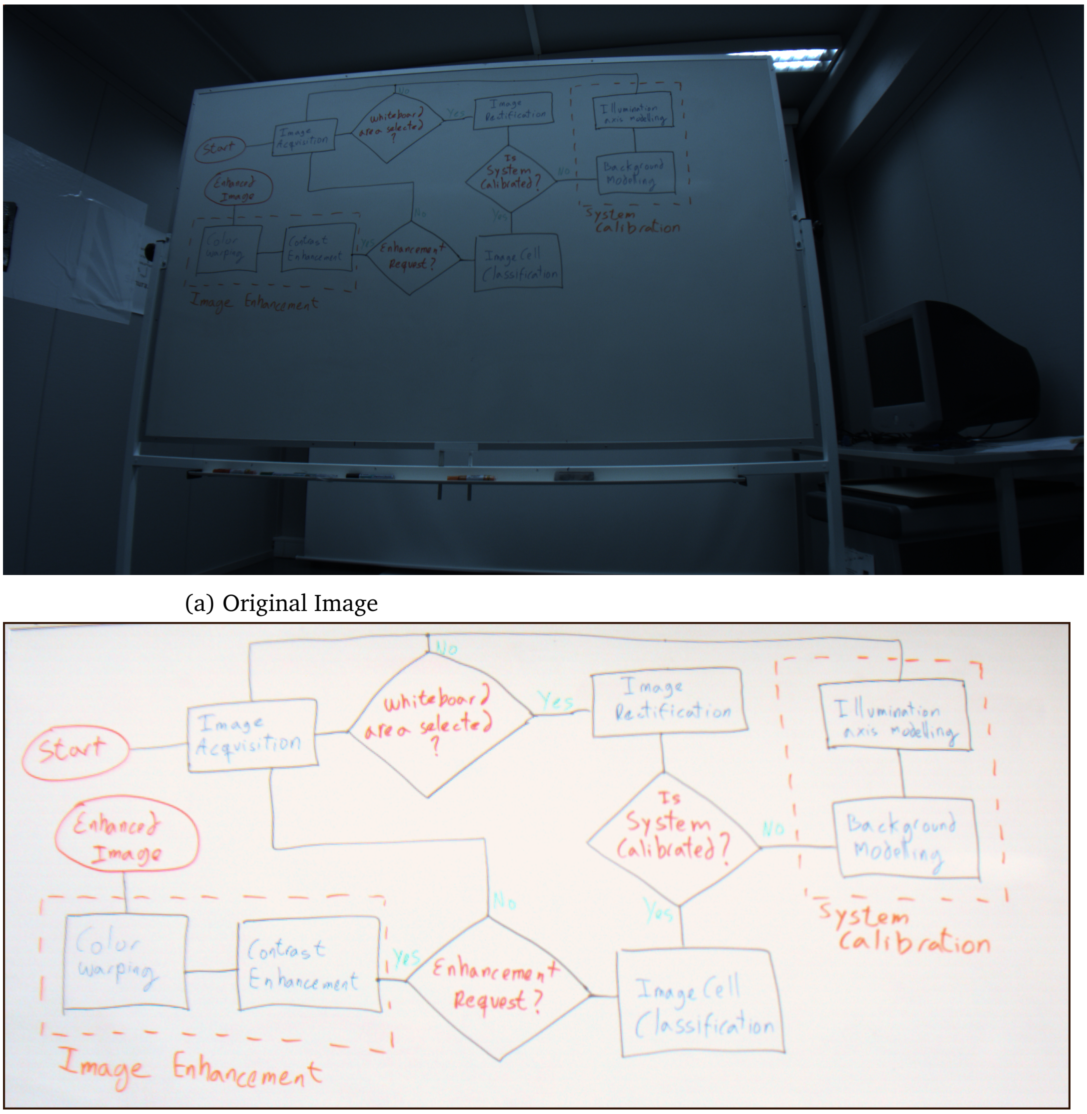
4.1.1 Camera

The camera used for this project was as a USB3 Vision Flea3 camera from the company Point Grey. More specifically it was a FL3-U3-88S2C-C. It is a high resolution camera (4096 x 2160) with a maximum frame rate of 21 fps. It uses a CMOS Sony IMX121 sensor with a pixel size of 1.55 µm. It is a small industrial digital “ice cube” camera measuring just 29x29x30 mm. The camera also provides on-camera processing including color interpolation, look up table, gamma correction, and pixel binning functionality [61].

One of the most interesting features of this camera is its Format_7 video custom mode. Format_7 modes are specifications that are not defined by DCAM (IIDC IEEE-1394-based Digital Camera Specification) but rather by the camera manufacturer. The camera Format_7 mode is called region of interest (ROI) or “sub-windowing” and allows the user to select a specific portion of an image for the camera to transmit. This is done by skipping CMOS rows until it reaches the region of interest and consequently send images at faster frame rates. ROI has no effect on color processing, since the Bayer tiling in the region of interest is left intact [62].

4.1.2 Software

The project was developed entirely in Python. We used the mathematical package NumPy for linear algebra operations and N-dimensional array manipulation. We used the library OpenCV version 3.0 beta for image processing and computer vision. For manipulating the Flea3 camera we used the Point Grey FlyCapture2 SDK and the python wrapper pyfly2 [64]. Furthermore we



(a) Original Image

(b) Enhanced Image

Figure 15: Diagram of the system architecture drawn on a whiteboard

Figure 16: Camera used for this project [63]

used the libraries pyfly_utils and video_tables developed by Kubicam [3] for white balancing and gamma correction. We also used an algorithm for peak detection for Section 4.5.2 [65]. Finally We used Matlab for analysis of data, testing and prototyping.

4.2 Image cropping and rectification

In the first step in our application, the user must click on the corners of the whiteboard. The software automatically crops the whiteboard area using a mask described by the area inside the whiteboard corners. However as one can see in Fig.15a because of perspective distortion the whiteboard area appears to be a trapezoid area. The next step is to transform this area into a rectangle. One approach is to use the perspective transform. The main reason to use this transformation over other mapping techniques is that perspective transformation maps straight lines to straight lines (Other similar transforms may introduce distortions that maps straight lines in one space to curves in another space) [66] [67]. The task becomes taking the coordinates of the four corners of the first quadrilateral (x_n and y_n) and the coordinates of the four corners of the new rectangle (X_n and Y_n) and compute the perspective transform that maps the pixels from the quadrilateral to the appropriate position on the rectangle. For that we need to estimate the size of the final image. In Fig.17 the lengths of the lateral sides of the original quadrangle are denoted by H_1 and H_2 and for the upper and lower sides are denoted W_1 and W_2. We determine the new rectangle width and height as:

$$\hat{W} = \max(W_1, W_2) \qquad \text{and} \qquad \hat{H} = \max(H_1, H_2) \tag{4.1}$$

Once the new size is determined the rectifying matrix **M** (Homography matrix) can be computed as:

$$\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{4.2}$$

This is equivalent to:

$$X = \frac{ax + by + c}{gx + hy + 1} \tag{4.3}$$

$$Y = \frac{dx + ey + f}{gx + hy + 1} \tag{4.4}$$

We can extend these expressions as:

$$\begin{aligned} X &= ax + by + c - gXx - hYy \\ Y &= dx + ey + f - gXx - hYy \end{aligned} \tag{4.5}$$

If we add some zero terms we get:

$$\begin{aligned} X &= ax + by + c + 0d + 0e + 0f - gXx - hYy \\ Y &= 0x + 0y + 0c + dx + ey + f - gXx - hYy \end{aligned} \tag{4.6}$$

So if we express the latter expression as the product of a matrix and a vector we get:

$$\begin{bmatrix} x & y & 1 & 0 & 0 & 0 & -Xx & -Yy \\ 0 & 0 & 0 & x & y & 1 & -Xx & -Yy \\ & & & & \vdots & & & \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \end{bmatrix} = \begin{bmatrix} X \\ Y \\ \vdots \end{bmatrix} \tag{4.7}$$

And we can now calculate the transformation matrix using 4 pair of points (the corners of both quadrilaterals):

$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -X_1x_1 & -Y_1y_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -X_1x_1 & -Y_1y_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -X_2x_2 & -Y_2y_2 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -X_2x_2 & -Y_2y_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n & y_n & 1 & 0 & 0 & 0 & -X_nx_n & -Y_ny_n \\ 0 & 0 & 0 & x_n & y_n & 1 & -X_nx_n & -Y_ny_n \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \end{bmatrix} = \begin{bmatrix} X_1 \\ Y_1 \\ X_2 \\ Y_2 \\ \vdots \\ X_n \\ Y_n \end{bmatrix} \tag{4.8}$$

The form of Equation 4.8 is $Az = b$ and it can be solved with a least squares estimation of the parameters of vector z that satisfies the linear relationship between A and B. We transform the input image (I) to the output image (v) using the calculated homography matrix $(\mathbf{M})$:

$$v(x,y) = I\left(\frac{M_{11}x + M_{12}y + M_{13}}{M_{31}x + M_{32}y + M_{33}}, \frac{M_{21}x + M_{22}y + M_{23}}{M_{31}x + M_{32}y + M_{33}}\right) \tag{4.9}$$

Finally we calculate the pixel values of our output image $u(x,y)$ using bilinear interpolation [68]:

$$u(x,y) = (1-\langle x\rangle)(1-\langle y\rangle)v_{\lfloor x\rfloor,\lfloor y\rfloor} + \langle x\rangle(1-\langle y\rangle)v_{\lfloor x\rfloor+1,\lfloor y\rfloor} + (1-\langle x\rangle)\langle y\rangle v_{\lfloor x\rfloor,\lfloor y\rfloor+1} + \langle x\rangle\langle y\rangle v_{\lfloor x\rfloor+1,\lfloor y\rfloor+1} \tag{4.10}$$

where $\lfloor x\rfloor$ denotes the floor function of x and $\langle x\rangle = x - \lfloor x\rfloor$.

4.3 System Calibration

4.3.1 Illumination Axis Estimation

The next task is to model the color distribution in the RGB color space. We follow a similar method as discussed in Section 3.2.3 and divide our image into small cells of 15×15 pixels (Fig.

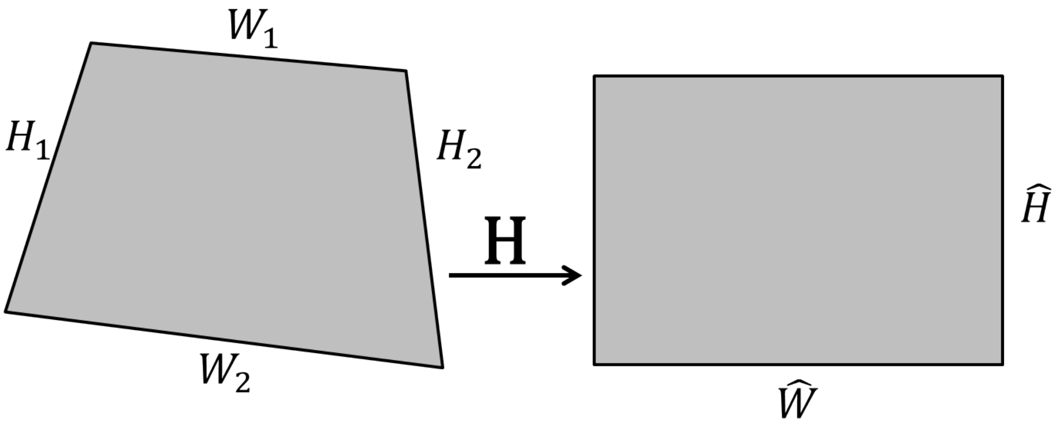


Figure 17: Rectification of an whiteboard image. (Left) original shape (Right) rectified shape

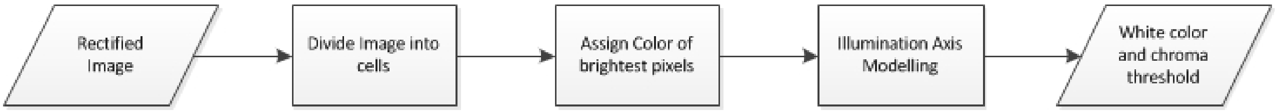

Rectified Image
Divide Image into cells
Assign Color of brightest pixels
Illumination Axis Modelling
White color and chroma threshold

Figure 18: Color modelling process

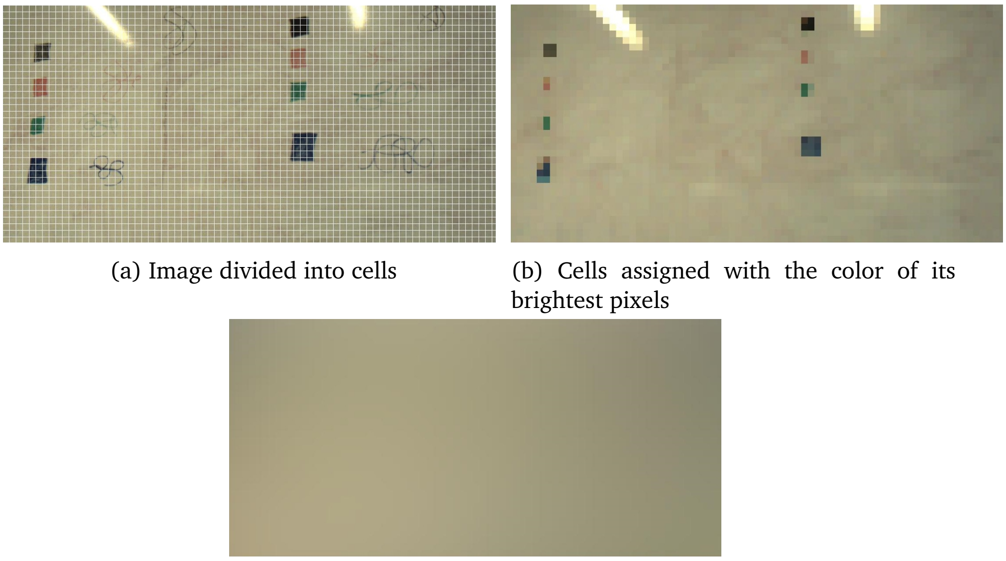
19a). Then we look for the 10% brightest pixels in the cell, calculate the color mean and assign this value as the color of the cell (Fig.19b). This step assures that we will get the whiteboard color of the cell even in presence of pen-strokes. The next step is to calculate the illumination axis as described in Section 3.3.1. One thing to be considered is the effect specular reflection might have in the color modelling. When the camera is capturing areas of a whiteboard with strong specular reflection the sensor might get saturated and deliver color values that highly deviate from the rest of the color distribution (outliers). We avoid these outliers by simply ignoring color which maximum value is close to maximum saturation ($max(I_r, I_g, I_b) \approx 1$).

The next step is to calculate a series of values that will be used as thresholds in other step of our system. First we calculate the distance of the image colors and illumination (Equation 3.25). Then we estimate the chroma threshold as the mean of 10% highest chroma values. This value will be used in Section 4.5.2 as threshold criteria for color sampling. Secondly we estimate the lightness of the cells w.r.t the illumination axis (Equation 3.23). Then we calculate the lightness threshold as the mean lightness which we will be used in Section 4.4 as classification criteria. We also calculate the estimated illuminant color W_e using Equation 3.20.

4.3.2 Background Modelling Implementation

The following task is to estimate the whiteboard image background. We follow the same steps as in the previous section and we divide the image into cells and assign the color of the brightest pixels. The next step is to use the extracted color data from the cells and fit a third degree polynomial surface to each individual RGB color channel using the method described in Section 3.2.4.

There are two drawbacks with this method. The first one comes from the subsampling method-



(a) Image divided into cells

(b) Cells assigned with the color of its brightest pixels

(c) Modelled Background

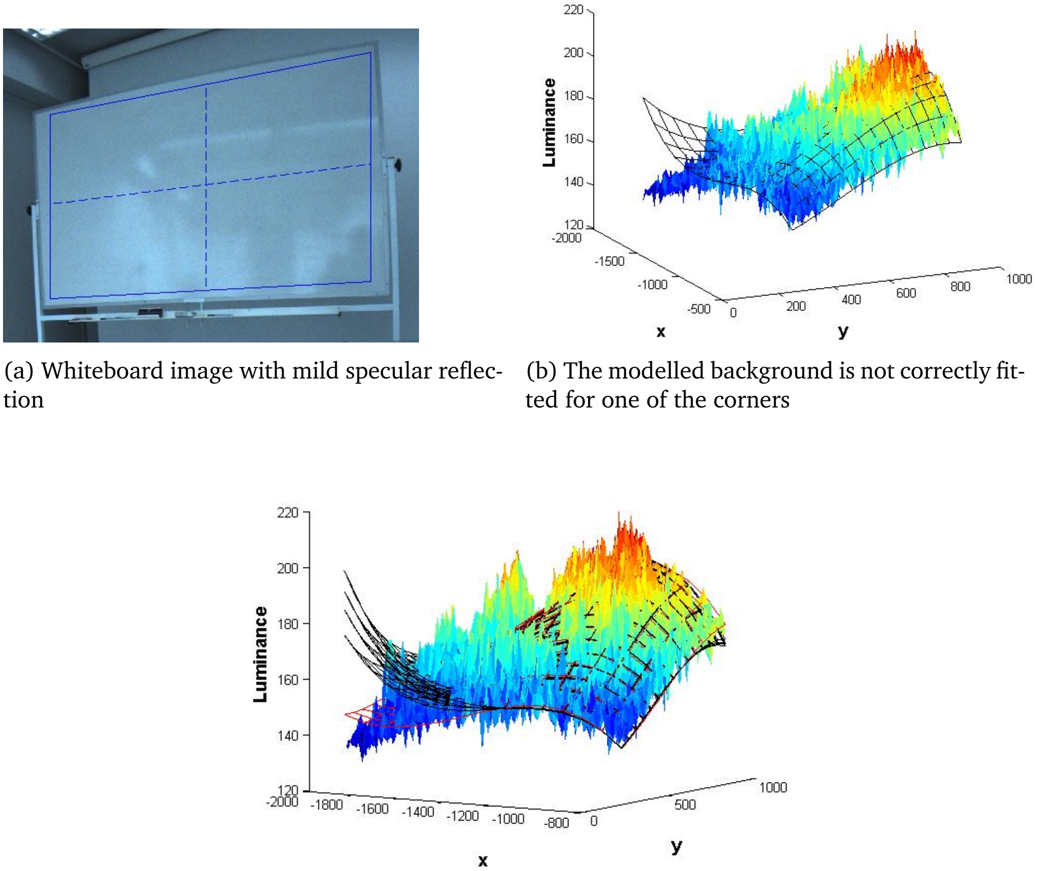
Figure 19: Background Modelling Scheme

ology. Since the subsamples used for modelling the surface are chosen randomly, two calculated surfaces will be similar but different from each other hurting its repeatability for experimental testing. The second drawback is that in certain scenarios (Normally in presence of some mild specular reflection) the estimated values are largely miscalculated in the corner areas. Consequently the estimated surface does not correctly fit the whiteboard luminance(Fig.20b).

We were able to solve this issue taking advantage of the first drawback. Since different iterations of the modelling scheme offer different results we improved our results by comparing the difference between the estimated surface and the measured value and if it was bigger than a certain threshold we forced our system to repeat the modelling with a new set of random samples until we found a surface that satisfied this threshold (Fig.20c). We can see the implemented solution more clearly in the flowchart in Fig.21. The estimated values are used to create background images as seen in Fig.19c.

4.4 Image Cell Classification

In previous sections we stated that we divided the image into small cells in order to lower the computational cost for the background and color modelling. However we can also use the cells in order to locally analyse the whiteboard image. Our first step is to identify if there are any foreground objects (A person writing on the whiteboard or an occluding object between the camera and the whiteboard) by classifying the cells as whiteboard or foreground. We do this by comparing the color distance of the image cell color ($I_{cell}(x,y)$) and luminance mean ($L_I(x,y)$)



(a) Whiteboard image with mild specular reflection

(b) The modelled background is not correctly fitted for one of the corners

(c) The black surfaces are rejected by our system. The red surface is the one accepted and used to model the background.

Figure 20: Corner over-estimation problem and solution

with its equivalent from the estimated background ($B_{cell}(x, y)$ and L_B respectively):

$$I_{cell}(x, y) = \begin{cases} Whiteboard & if \quad \sqrt{(I_{cell} - B_{cell})^2} \leq u \quad and \quad minL \leq |L_I - L_B| \leq maxL \\ Foreground & otherwise \end{cases} \tag{4.11}$$

We can reclassify foreground cells by representing them as pixels in a binary image and we look for connected components. Foreground border cells are reclassified by performing a morphological dilation and we apply a flood fill algorithm to avoid the appearance of misclassified isolated whiteboard cells surrounded by foreground cells. Furthermore we analyze the foreground connected components and reclassify them as whiteboard if they are considered too small to be

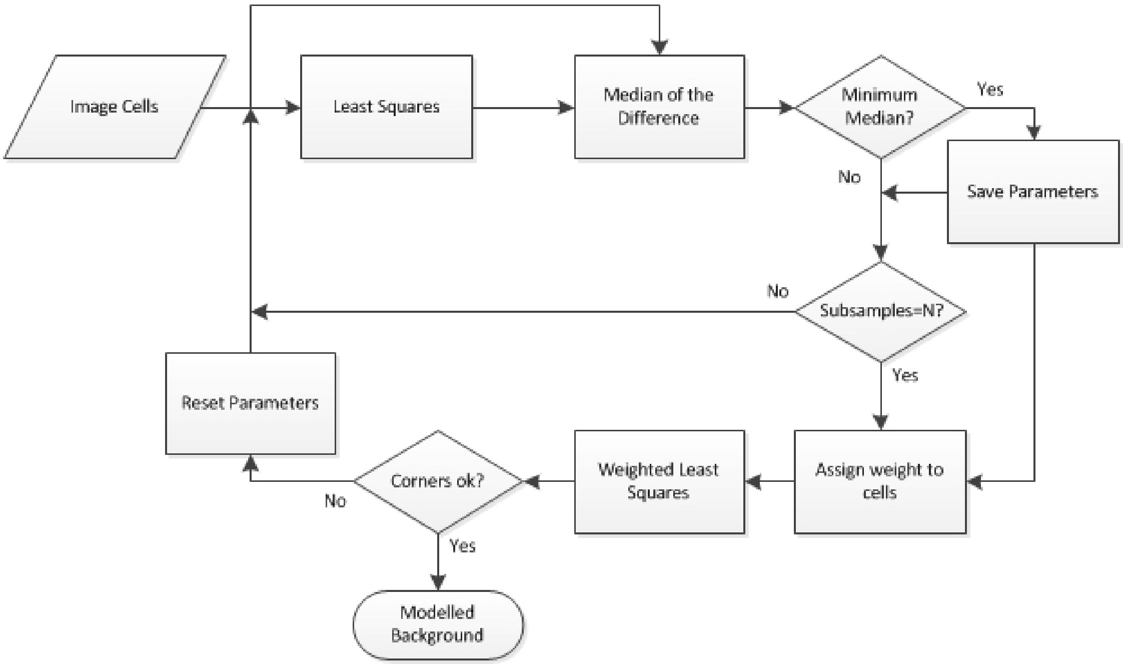


Figure 21: Background modelling process

an occlusion[1]. We get as result a foreground mask image that separates the whiteboard from occlusions (The whole cycle can be observed in Fig.22).

The next step is to classify the whiteboard cells as background and pen-strokes. We can assume that whiteboard cells have an uniform color (grey or white) and pen-stroke cells are mostly white or gray with some color mixed in. So we can classify the cells by measuring the standard deviation of its color distribution and comparing it with a threshold. However the standard deviation of a pen-stroke cell showed to be larger in areas with bigger luminance. Therefore we establish two thresholds depending on the cell lightness:

$$I_{cell}(x,y) = \begin{cases} Penstroke & if \quad \sigma_{cell} > T_{\sigma 1} \quad or \quad (\sigma_{cell} > T_{\sigma 2} \quad and \quad \Delta L_{cell} \geq 15) \\ Background & otherwise \end{cases} \tag{4.12}$$

and

$$\sigma_{cell} = max\left(\sigma_R, \sigma_G, \sigma_B\right) \tag{4.13}$$

where $sigma_{cell}$ is the maximum standard deviation of the color values in a cell for any of the RGB channels, $T_{\sigma 1}$ and $T_{\sigma 1}$ are the thresholds and ΔL_{cell} is the difference of luminance between the image cell and its equivalent estimated background. For our implementation $T_{\sigma 1} = 3$, $T_{\sigma 2} = 2$. This method however may become unstable due to camera noise. The color standard

[1] One example would be whiteboard filled drawings like the color rectangles in Fig.19b that might be misclassified as foreground

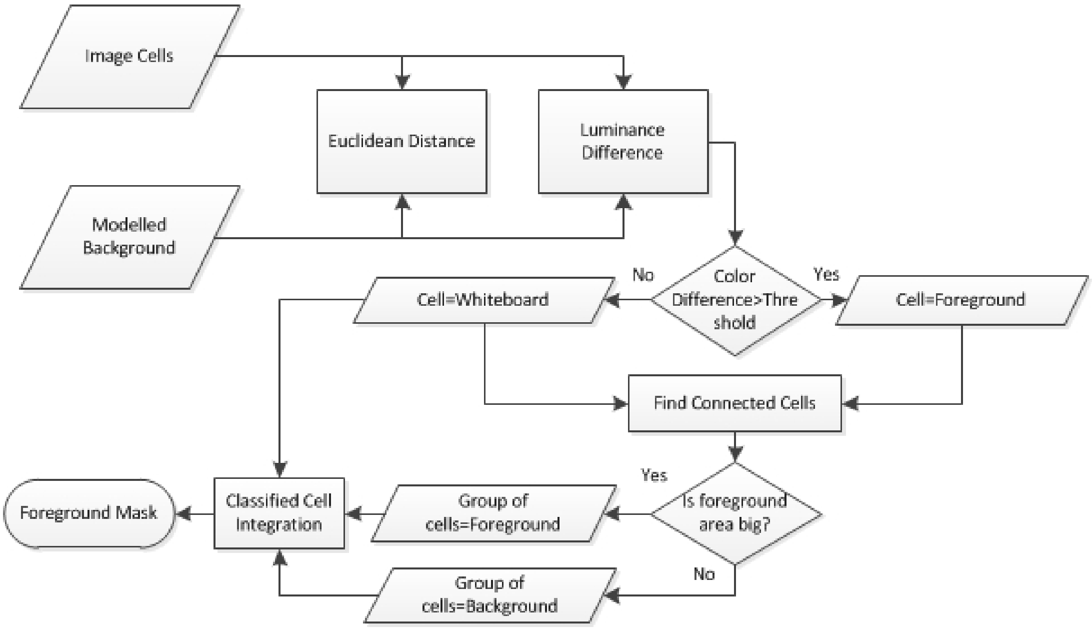


Figure 22: Foreground classification process

deviation of a white cell which in theory should have a small value might increase in presence of noise thus increasing the risk of being misclassified as a pen-stroke cell. Classical methods for noise reduction like Gaussian filtering might blur the pen-stroke areas and edge-preserving filtering methods like bilateral filter [69] or anisotropic diffusion [70] might be too computationally heavy for our video-conferencing system. Fortunately we can take advantage of the video-streaming of our video-conferencing system. Since the Kubicam system is intended to work with a static camera we can apply a simple recursive temporal filter to the whiteboard cells. First we store the first frame as the initial image cell (We assume that when we start our system there are no occlusions). Then we simply take the weighted average of the current image cell and the stored image cell frame. We use the foreground mask to check which cells were classified as whiteboard or as foreground. In order to avoid to contaminate our temporal filter with frames containing occlusion we state that formerly occluded cells must wait 5 iterations before we update their temporal filter.

$$I_{stored}(x,y) = \begin{cases} \alpha I_n(x,y) + (1-\alpha) I_{stored}(x,y) & \text{if} \quad t_m > 5 \\ I_{stored}(x,y) & \text{otherwise} \end{cases} \tag{4.14}$$

where $\alpha = 0.2$, n is the current frame and t_m is the number of consecutive iterations that one cell has been classified as part of the whiteboard. We can also see the results of the image cell classification in Fig.24b. The white cells represent pen-stroke cells, the black cells represent the background, the red cells represent occlusion and the blue cells are whiteboard cells which were recently ($t_m \leq 5$) classified as foreground.

Since the stored image cells are only updated when they are not occluded we can remove the occlusions free by replacing the occluded cells in our whiteboard image with the stored image

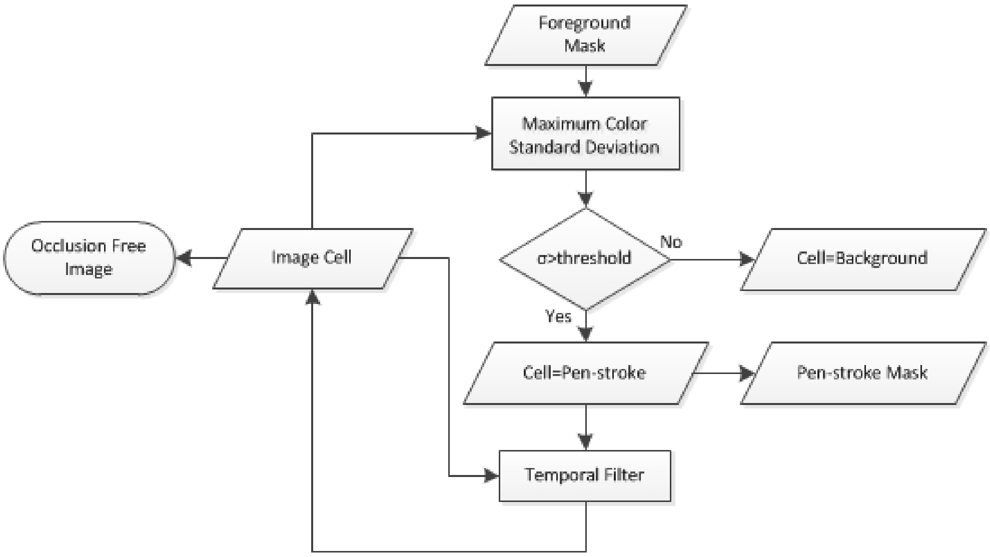


Figure 23: Whiteboard cell classification and temporal filtering process

cells (Fig.24c). The whole image cell classification between pen stroke and background, temporal filter and occlusion removal process can be seen in Fig.23.

4.5 Image Enhancement

This is the final step of our system and it's divided into 3 parts. First the image is whitened by rotating its colors from the illumination axis to the neutral gray axis and scaling the image with the background. The second step is to extract and identify the pen stroke color values. Finally the image is color warped in order to enhance the saturation and lightness of the pen strokes. The whole process can be seen in Fig.25. Furthermore we propose an alternative enhancement method that focus more on enhancing the image legibility.

4.5.1 Contrast Enhancement

The first step is to normalize the whiteboard colors by aligning them around the neutral gray axis. We base our approach on a technique called color cluster rotation [57] which creates a rotation matrix using the illumination axis $\overrightarrow{\mathbf{e_i}}$ (Equation 3.22) and the neutral gray axis $\left(\overrightarrow{w_i} = \frac{\sqrt{3}}{3}\{1,1,1\}\right)$. First we calculate the normal vector between the 2 axes:

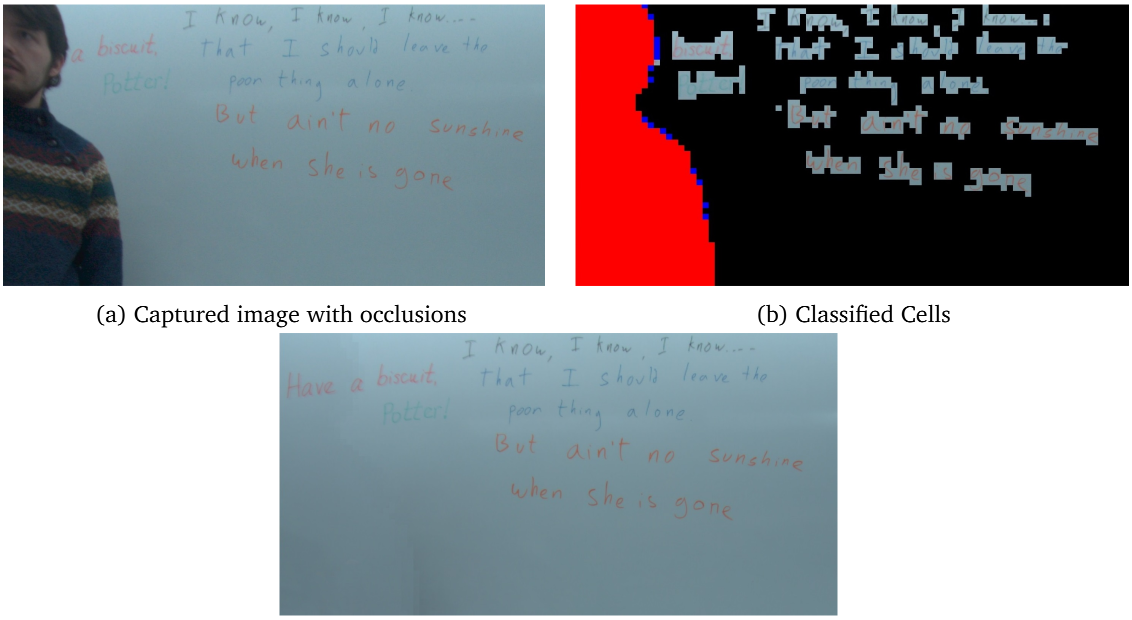
$$\overrightarrow{\mathbf{k}'} = \overrightarrow{w_i} \times \overrightarrow{\mathbf{e_i}} \tag{4.15}$$

where × denotes the cross product. We normalize $\overrightarrow{\mathbf{k}'}$ into an unitary vector by:

$$\overrightarrow{\mathbf{k}} = \frac{\overrightarrow{\mathbf{k}'}}{\|\overrightarrow{\mathbf{k}'}\|} \tag{4.16}$$

Then we calculate the rotation angle ϕ from the dot product of the illumination axis and the neutral gray axis:

$$\cos\phi = \overrightarrow{w_i} \bullet \overrightarrow{\mathbf{e_i}} \tag{4.17}$$



(a) Captured image with occlusions

(b) Classified Cells

(c) Occlusion removed from image

Figure 24: Image cell classification

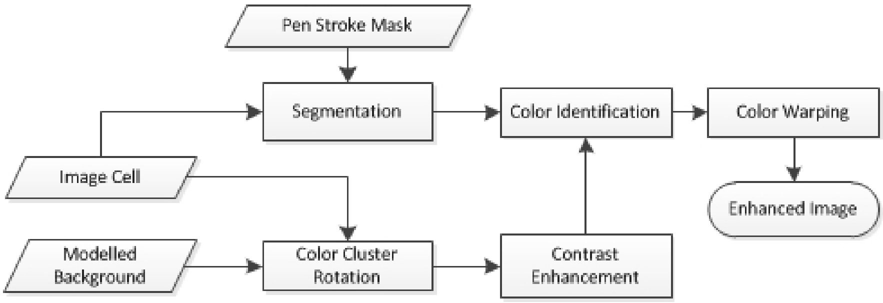
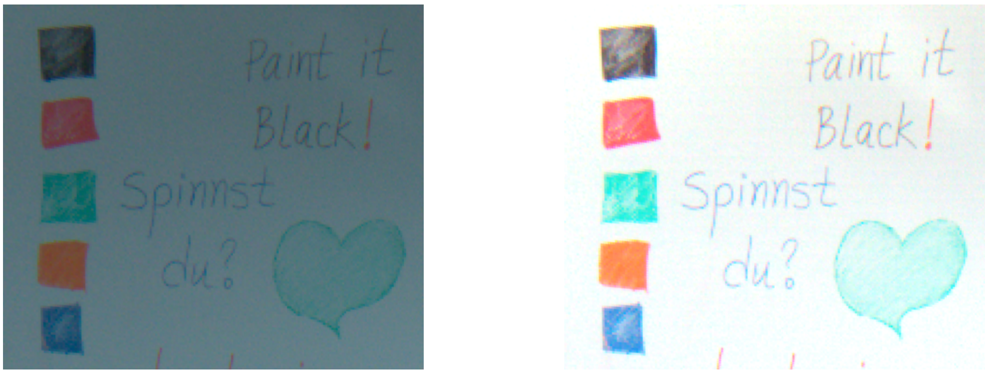


Figure 25: Whiteboard image enhancement process



(a) Original Image

(b) Output Image

Figure 26: Contrast Enhancement Results

Now we calculate the rotation matrix by using Rodrigues formula [71] for an angle ϕ around an axis expressed as a vector $\overrightarrow{\mathbf{k}}$:

$$R_3(\phi, \mathbf{k}) = \mathrm{Id}_3 - (\sin\phi)\mathbf{K} + (1 - \cos\phi)\mathbf{K}^2 \tag{4.18}$$

where Id_3 is the 3×3 identity matrix and $\mathbf{K}$ is the cross-product matrix for the normal vector $\overrightarrow{\mathbf{k}} = (k_x, k_y, k_z)$:

$$\mathbf{K} = \frac{\sqrt{3}}{3} \begin{bmatrix} 0 & -k_z & k_y \\ k_z & 0 & -k_x \\ -k_y & k_x & 0 \end{bmatrix} \tag{4.19}$$

and $\mathbf{K}^2 = \mathbf{k}\mathbf{k}^T - \mathrm{Id}_3$. The next step is to convert each pixel Col to rotated pixel Col'. First we shift the colors using Bl_e so the illumination axis starts at the origin then we rotate using R_3:

$$Col' = R_3(\phi, \mathbf{K}) \bullet (Col - Bl_e) \tag{4.20}$$

We rotate the occlusion free image (Col') and the estimated background (F'). Then we try to make the whiteboard uniformly white by scaling each cell of the occlusion free image with its respective estimated background cell[2]:

$$Col_w(x, y) = \min\left(1, \frac{Col'(x, y)}{F'(x, y)}\right) \tag{4.21}$$

We can see the results in Fig.26. The background is made whiter and the contrast is enhanced. However some of the pen-strokes now have small saturation. We will address this problem in the following section.

4.5.2 Pen Stroke Color Identification

The next challenge is to identify the colors of the pen-strokes on the whiteboard. Depending on whiteboard user preference and dry-erase markers availability one whiteboard might have pen

[2] It is worth noting that this step is based on the assumption that we only need to consider the multiplicative element S_M of the shading model in order to obtain a shading free image (Equation 3.4)

strokes of one single color or different colors. As stated in Section 3.1.2 markers come in a great variety of colors, therefore we must develop a method which is able to identify which colors are present on the whiteboard without any a priori information.

First we take the foreground mask computed in Section 4.4 to identify which cells were classified as pen-strokes and we segment the pen-stroke in each cell using Otsu's thresholding method [14]. We refine the segmentation process by ignoring connected components that are smaller than one cell (A normal pen-stroke normally covers at least two consecutive cells). This process creates a mask that we apply to the resulting image from the previous section in order to extract only pen-stroke pixels.

Secondly we identify saturated colors. We can assume that any pen-stroke color except black have a high level of chroma. We calculate the chroma of the segmented colors using Equation 3.25. We select the pixels which chroma is larger than the chroma threshold (T_{ch}) calculated in Section 4.3.1.

$$Col'_e(x,y) = \begin{cases} \text{Selected} & \text{if} \quad C_e(x,y) \geq T_{ch} \\ \text{Not Selected} & \text{otherwise} \end{cases} \tag{4.22}$$

If no colors are selected we can assume that there are no saturated colored pen-strokes on the whiteboard image (Only black pen-strokes). The selected colors are then transformed into the rg-chromaticity space using Equation 3.27 and we calculate the selected color hue using Equation 3.29[3]. The next step is to look into the hue distribution in the selected colors and look for large concentration of hue angles. We can represent our hue data using an angular histogram plot. Let us take the whiteboard image from Section 4.3.1, segment the pen-strokes and select the saturated colors. In Fig.27a we can see the angular histogram of the hue distribution of the colors represented in the rg-chromaticity space of Fig.19a. We can unfold the angular histogram into a classical histogram (Fig.27b)and we apply an algorithm to detect any peaks [65]. There is a risk for this approach though. If there is a peak in the Hue angle near 0° this algorithm might ignore it[4]. As we can see from our example there are 3 peaks in our histogram which correspond to 3 color hues on the whiteboard (red,green and blue). Finally for each identified penstroke hue we look for selected colors which hue are closer to the identified color hues and select the one with largest chroma:

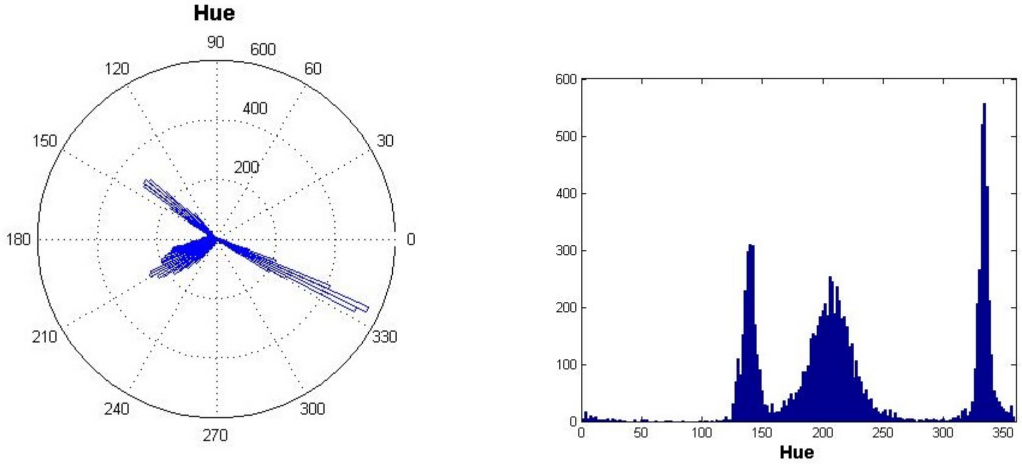
$$SelCol(i) = Col(j) \quad \text{if} \quad Ce(j) = \max(C_e) \quad \text{and} \quad \|H_e(i) - H_e(j)\| \leq Ang_{max}(i) \tag{4.23}$$

where i is number of identified color hues, $Col(j)$ is number of candidate pen-stroke color and $Ang_{max}(i)$ is the maximum hue difference (This difference is calculated as the maximum difference between an identified color and the 10% closest color hues).

Thirdly we identify whether there are any black pen-strokes on the whiteboard. We can assume that black pen-strokes will be less saturated and less bright that the pen-stroke of any other color marker. For these reasons we calculate both the chroma and lightness of the segmented colors using Equation 3.25 and Equation 3.23. We select the pixels which chroma is lesser than the

[3] It's worth noting that since we rotated the whiteboard colors from the illumination axis to the neutral gray axis in Section 4.5.1 the white value W_e now corresponds to the neutral white color ($r_{w'} = \frac{1}{3}$ and $g_{w'} = \frac{1}{3}$)

[4] We have observed that there is a shift in our camera color wheel and that this area will correspond to a reddish magenta. Fortunately reddish magenta is not a common color for dry-erase markers. Furthermore if we simply shift the angles 60°we should be able to find the peak in the shifted region



(a) Angular Histogram

(b) Classic Histogram

Figure 27: Hue Histograms

chroma threshold calculated in Section 4.3.1.

$$Col''_e(x,y)=\begin{cases}Selected & if \quad C_e(x,y)\le T_{ch}\\ Not\ Selected & otherwise\end{cases} \tag{4.24}$$

From this selection we extract the colors which have the smallest chroma and least lightness.

$$SelBl = Col(j) \quad if \quad Le(j)=\min\left(L_e\right) \quad and \quad C_e(j)\le Chro_{max}(i) \tag{4.25}$$

where $SelBl$ is the identified color and $Chro_{max}(i)$ is the maximum black chroma (This value is calculated as the maximum chroma of the 10% color values with smallest chroma). Although the identified color will have low chroma there is no guarantee that it is black and not a low chroma version of one of the previously identified saturated colors. In order to confirm that it is in fact black we must check that it has a lower lightness than the other identified colors.

$$SelBl=\begin{cases}Black & if \quad L_{SelBl}<L_{SelCol}(i) \quad \forall i\\ Not\ Black & otherwise\end{cases} \tag{4.26}$$

4.5.3 Color Warping

The final step is to enhance the pen-stroke color lightness and saturation. In order to avoid creating unnatural looking images we need a method that can be applied all over the image. We can use a method that, given a set of target colors, will be able to transform the color distribution in a color space. We use a technique called color warping [72]. This technique is based on image warping in which an image is geometrically deformed based on a set of source/destination pixels in the image plane. The main difference is that color warping transforms a set of source/destination color points in a given color space. This set of color pairs define the warping of the color space according to the following properties:

- The source color is directly mapped to the destination color.

- Colors close to a given source color end up close to the corresponding destination color.
- Colors that have the same distance to two source colors are influenced equally by the two source/destination pairs
- Colors are influenced more by closer source colors.

In the previous section we described how to obtain the source colors. We calculate the destination colors based on the vectorial representation from Equation 3.26. From each identified color we calculate $\overrightarrow{L_e}(i)$ and $\overrightarrow{C_e}(i)$ (Equation 3.23 and Equation 3.25). We can represent this vector as a multiplication of the unit vector and a magnitude so:

$$\overrightarrow{L_e}(i) = Le(i)\overrightarrow{e_i} \qquad \text{and} \qquad \overrightarrow{C_e}(i) = Ce(i)\overrightarrow{c_i} \tag{4.27}$$

where L_e and C_e are the color lightness and chroma respectively, $\overrightarrow{e_i}$ is the normalized projected vector (it can be either the illumination axis or in our case the neutral gray axis) and $\overrightarrow{c_i}$ is the normalized rejected vector. Now we can create our new destination colors by changing the values of chroma and lightness. Our new colors are:

$$Col_D(i) = \beta_1 Le(i)\overrightarrow{e_i} + \beta_2 Ce(i)\overrightarrow{c_i} \tag{4.28}$$

For the saturated colors we want a destination color with increased saturation and decreased lightness, thus $\beta_1 = 0.75$ and $\beta_2 = 3$. For the black color we want a destination color with decreased lightness and minimum saturation thus $\beta_1 = 0.75$ and $\beta_2 = 0$. We will include one last source/destination color pair. The source will be the the color in the estimated background with maximum lightness and the destination color neutral white ($White = (1, 1, 1)$). This last colour pair is introduced in order to counteract the effect of color warping the background pixels towards saturated and black source/destination colors (Since we want them to remain white). The final step is to warp the image in the RGB color space. Given N source color values $C_s i, i = 1, 2, \dots, N$, in a given color space with their corresponding destination color values $C_D i, i = 1, 2, \dots, N$, the change of a color is calculated as the weighted sum of the contributions of all source and destination pairs [72]:

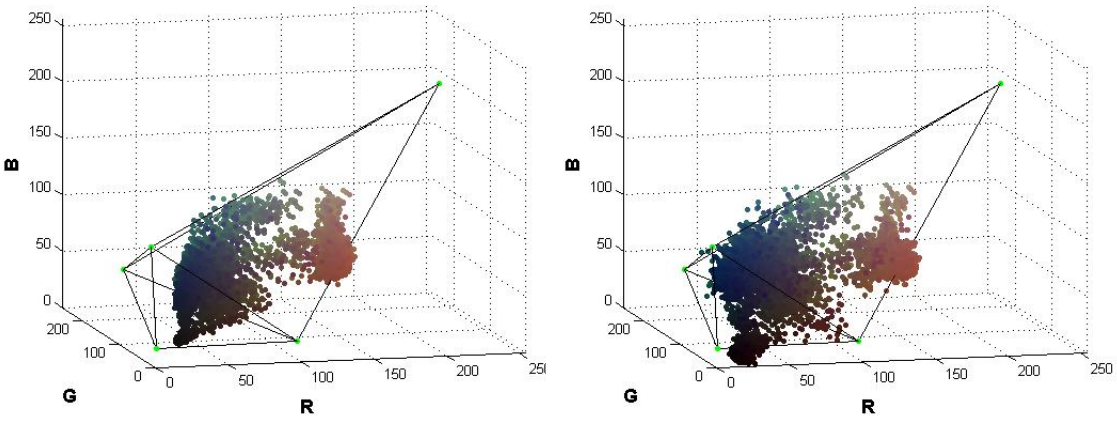
$$C_O = C_I + \sum_{i=1}^{N} \omega_1(i)\omega_2(i)\left(C_D(i) - C_S(i)\right) \tag{4.29}$$

where C_I is the input color,C_O the warped color and $d(i) = \|C_I - C_S(i)\|$ is the euclidean distance between the input color and the ith source color. The first weight function, $\omega_1(i)$, is calculated as a normalized inverse distance between the input color and the ith source color:

$$\omega_1(i) = \begin{cases} \frac{1/d(i)}{\sum_{j=1}^{N} 1/d(j)} & \text{if} \quad \min d(i) > 0 \\ \delta(i, i_{min}) & \text{if} \quad \min d(i) = 0 \end{cases} \tag{4.30}$$

where $i_{min} = \arg\min d(i)$ and $delta(i, i_{min})$ is the Kronecker delta function in which the result is 1 if $i = i_{min}$ and 0 otherwise. The second weight function $\omega_2(i)$ is an exponential function of the distance:

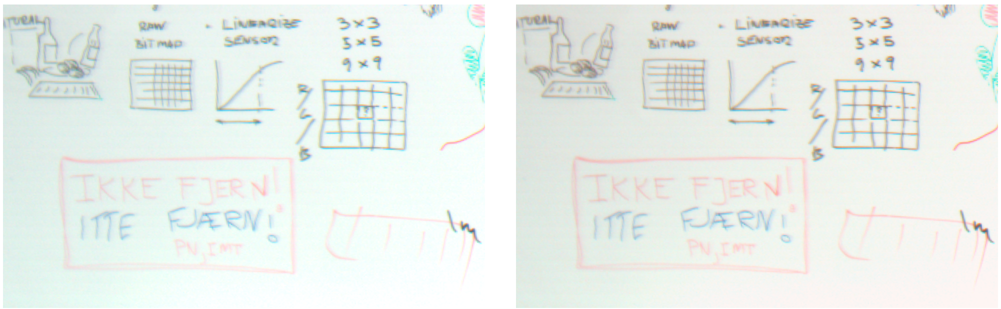
$$\omega_2(i) = e^{\frac{-d^2(i)}{2\sigma^2}} \tag{4.31}$$



(a) Input Colors

(b) Colors warped in the RGB color space

Figure 28: Color Warping in the RGB color space



(a) Whitened Image

(b) Enhanced Image

Figure 29: Color Warping Results

corresponding to a normal distribution with a standard deviation σ. This weight function reduces the influence of the warping algorithms for colors distant from the source. In Fig.28 we can see how the color values move towards the new destination (The green points) in the RGB color space. In Fig.29 we can see the the resulting image has pen-strokes with more saturated colors and a more uniform background. More especifically in Fig.29a we can see how weak the red lines are and their enhancement in Fig.29b.

4.6 Alternative Enhancement Method

As shown in the examples in Fig.29 we obtain enhanced images with better contrast and more saturated pen-stroke colors. However, there is a compromise between how much enhancement is done to make the image more visually pleasant (Appearance) and how much is enhanced to make the written text in the whiteboard easier to read (Legibility)[5]. When the captured pen-

[5] Legibility is sometimes confused with readabilty. Legibility can be defined as the ability a human reader to read something without effort. Readability can be defined not on a letter by letter basis, but how the combination of letter

strokes have a low contrast w.r.t. the whiteboard our enhancement step might remove some of its details. This dilemma is more clearly illustrated in the case of green dry-erase markers. During the development of this project we observed that green markers ink pen strokes were less saturated and have lower contrast w.r.t. the background than any other type of marker. Let's take input image Fig.30a and enhance with our proposed method (Let us refer to it as Method A) as shown in Fig.30b. As we can see the overall image contrast was increased and the pen stroke colors look more saturated but the words written in green ink are difficult to read. A careful analysis of our process have showed us that during the color cluster rotation step (Section 4.5.1) our green pen-strokes turn into a light greenish cyan color with low saturation.

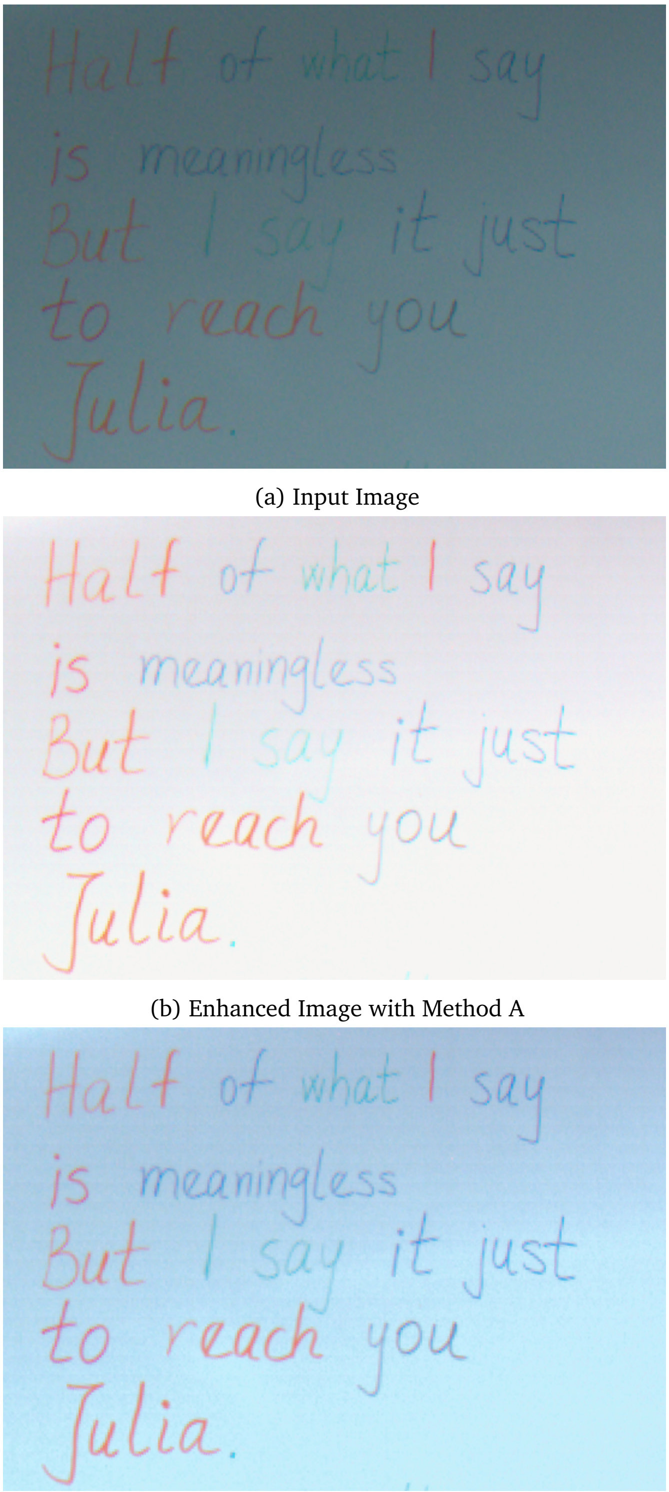
Let's consider an alternative approach (Let us refer to it as Method B). Instead of rotating the illumination axis we try to enhance the contrast along the current axis position. We enhance the contrast using a modified version of Equation 4.21 where we try to make the background uniform and give it the color of the illuminant (Calculated with Equation 3.20).

$$\mathrm{Col}_w(x,y,i) = \min_{i=R,G,B}\left(W_e(i), W_e(i)\frac{\mathrm{Col}(x,y,i)}{F(x,y,i)}\right) \tag{4.32}$$

The rest of the enhancement is made exactly as proposed in Section 4.5.2 and Section 4.5.3[6]. As we can see in Fig.30c the words are more legible in Method B compared to Method A (Fig.30b) however it has a lower global contrast. In the next chapter we will compare in more detail the pros and cons of Method A and Method B.

are read within a larger body of text. In other words, readability is defined by the amount of effort one needs to make to read text, not single characters [73].

[6] When selecting the destination colors for color warping, the last destination color is the color of the illuminant (W_e) and not neutral white ($W = (1,1,1)$).



(a) Input Image

(b) Enhanced Image with Method A

(c) Enhanced Image with Method B

Figure 30: Enhancement Methods Comparison

5 Results

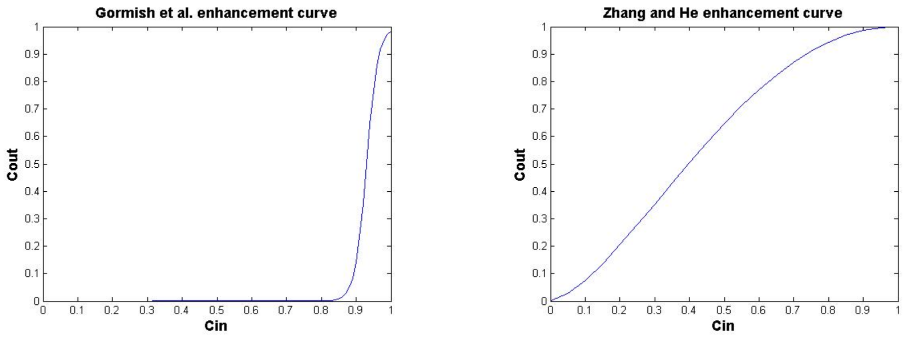
5.1 State of the Art comparison

In order to test the performance of our proposed methods we decide to compare them with other state of the art whiteboard content enhancement methods (Section 2.2). We have decided to test both our original proposed enhancement method (We will refer to it as Method A) and the alternative enhancement method proposed in Section 4.6 (We will refer to it as Method B). The state of the art methods selected for comparison were the one proposed by Gormish et al. [19] and the one proposed by Zhang and He [21][1]. We implemented both methods using the parameters as specified in their respective papers ($C = 60$ and $k = 0.93$ for Equation 2.1 and $p = 0.75$ for Equation 2.2). We can do an early prediction of the enhancement results of these two methods by just looking at their respective enhancement curves. Since the method of Gormish et al. is only applied to their segmented pen-strokes we will see images in which the color of the pen strokes will be shifted to its closer primary color (Fig.31a) and the background is turned to neutral white. On the other hand, since Zhang and He divides the input image with their modelled background, their enhancement curve will reduce background image noise and try to enhance the pen strokes color saturation.

5.1.1 Color Appearance Comparison

At first glance each method seem to show different results from each other. Let's take a first set of image results (Fig.32). Method A increased the contrast but the background turned into a reddish white(An undesirable effect from color warping). Method B preserve the colors better but its

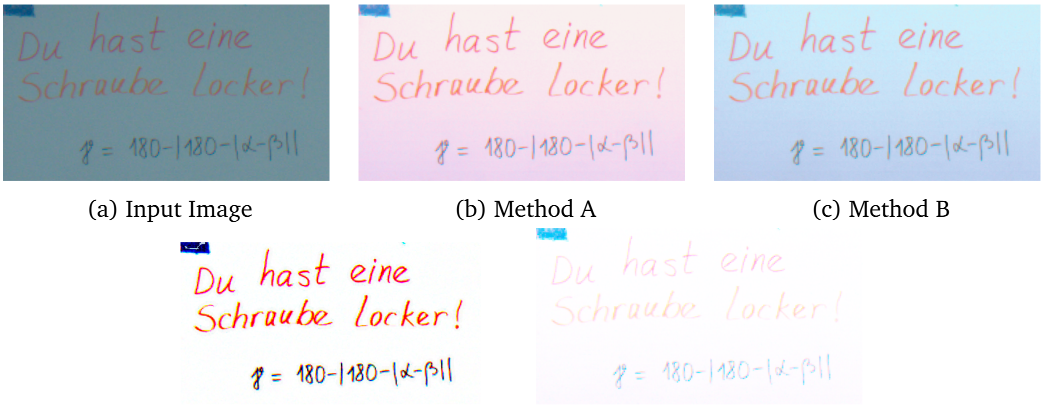
[1] We decided to omit the method proposed by Sakshuwong and Tsai [13] because they don't provide enough information about the parameters in their paper and the one proposed by He et al. [16] since it only tries to diffuse the specular reflection but doesn't try to enhance the contrast of the pen-strokes



(a) Gormish et al.

(b) Zhang and He

Figure 31: Enhancement Curves



(a) Input Image

(b) Method A

(c) Method B

(d) Gormish et al.

(e) Zhang and He

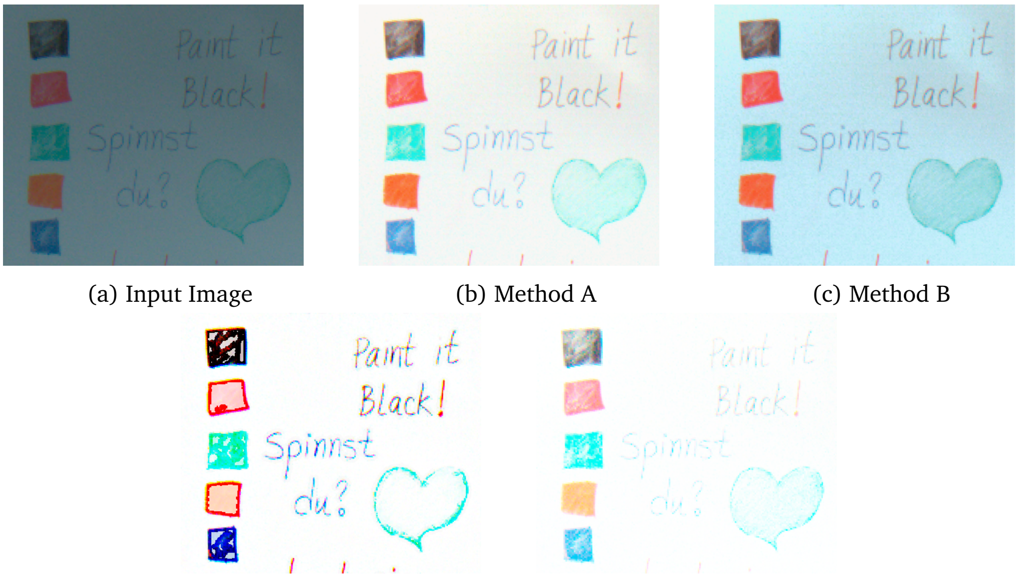
Figure 32: Enhancement Results: Image set 1

background is the darkest one. The method from Gormish shows to have a brighter background and bigger saturation compared to the other methods but the orange text has changed to red. The method from Zhang and He has a bright background but the colour saturation of the pen-strokes have been reduced and it's the image with less contrast.

Let's take a second set of image results (Fig.33). Both Method A and Method B enhance the color of the pen-strokes although Method A has a brighter background and a better contrast between background and pen strokes. However the method by Gormish et al. shows some odd results since only certain areas were enhanced in the color filled squares and some others were bleached resulting in the appearance of "white holes". This happens because this method relies heavily on its segmentation method to decide which areas it enhances (Since it uses canny edge detector it will only enhance the border areas). Once again the method by Zhang and He shows a bright background but a low contrast between background and pen strokes.

Let us give a closer look at the pen stroke color values. First we examine how the black pen-strokes are affected by these enhancement methods. We zoom on the image from Fig.33 and crop the word "it" (Fig.34). As we can see Method A mostly preserves the pen-stroke gray color with some minor blue speckles. Method B shows a similar behavior as method A but the blue speckles are less noticeable because of the bluish background. The method by Zhang and He have made the pen-stroke lighter and the blue speckles are more noticeable. The method from Gormish et al. shows a random color variation between the pen stroke pixels and a white halo around the letter[2]. The random color pixels can be explained by the enhancement curve in Fig.31a and Equation2.1. In Gormish et al. method a gray pixel would be transformed into a black value if its color value is below the enhancement threshold in the three RGB image channels, however when the value of one or two of the channels is above this threshold the pixel is transformed into a saturated color. The halo effect can be explained by the segmentation step since, as mentioned in Section 2.2, the segmented pen strokes are morphologically dilated to avoid having double

[2] You might have to adjust the brightness and contrast of your screen or move the angle of your laptop screen to see this effect since it's not noticeable at first glance



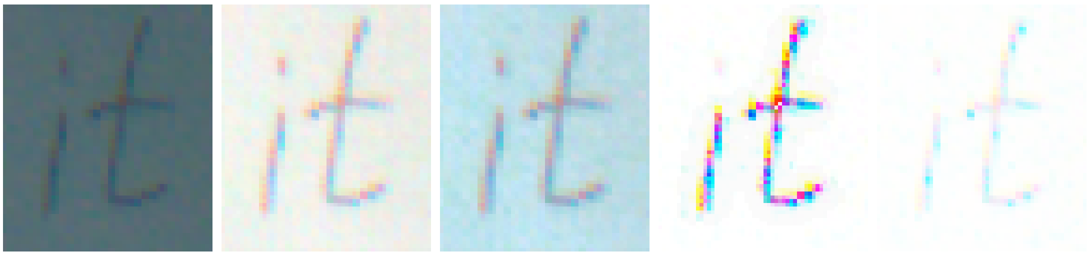
(a) Input Image (b) Method A (c) Method B

(d) Gormish et al. (e) Zhang and He

Figure 33: Enhancement Results: Image set 2

edged pen-stroke segments, thus the segmentation mask ends up covering part of the background area around the pen strokes (The resulting enhanced color of this area will be very light but not exactly pure white $W = [1, 1, 1]$ as the background).

Now let us a have a closer look at pen strokes with saturated colors. We take the pie-chart image in Fig.35a. We zoom the image and crop the center section of it (Fig.35b). As we can see Method A give us more saturated colors. Method B give us similar results but the colors are less saturated than in Method A. Gormish et al. give us the most saturated colors but it has white holes and some colors been completely changed (For instance the orange section now has red and yellow patches and the green section has now green and blue patches). The method of Zhang and He gave us lighter colors with less saturation than the original ones.



(a) Input Image (b) Method A (c) Method B (d) Gormish et al. (e) Zhang and He

Figure 34: Enhancement Results: Black pen strokes

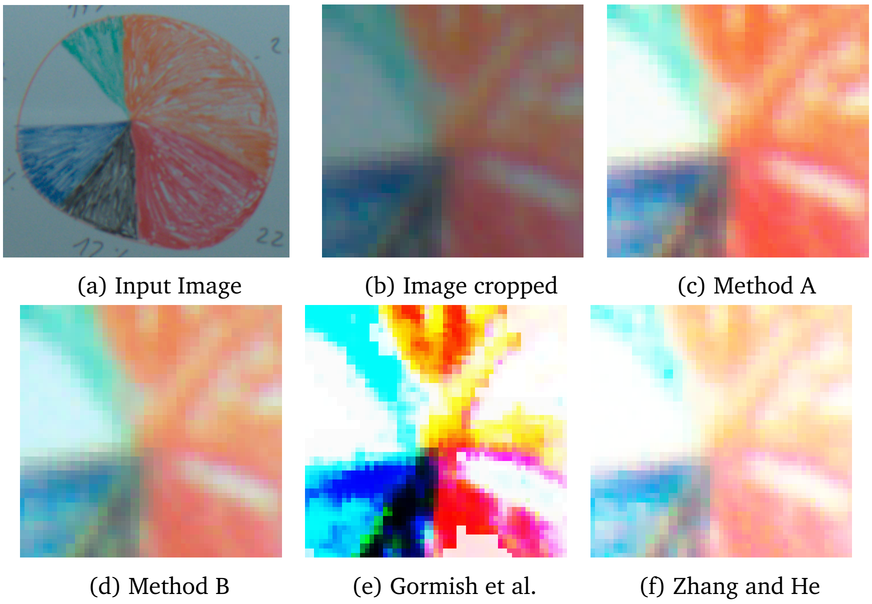

(a) Input Image
(b) Image cropped
(c) Method A
(d) Method B
(e) Gormish et al.
(f) Zhang and He

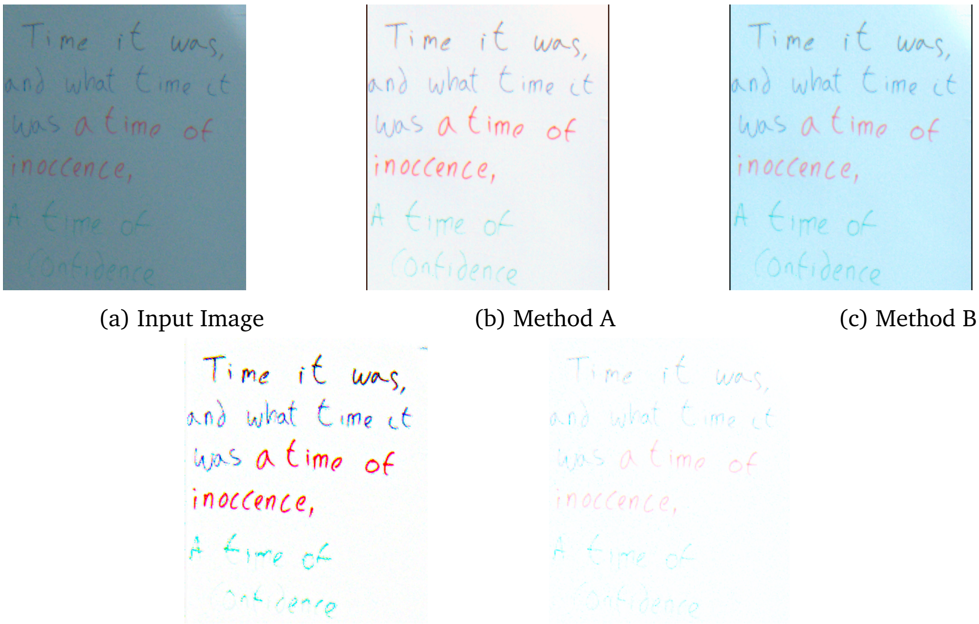
Figure 35: Enhancement Results: saturated color pen strokes

5.1.2 Legibility Comparison

Now let us evaluate the images legibility. First we can check again the enhanced images in Fig.32. Since we can see all the details in the letters in all the images the image produced by the method of Gormish et al. seems to be the most legible since it has the strongest contrast and the one produced by Zhang and He is the least legible due to the low contrast. However when we test the enhancement methods in another image set (Fig.36) we can see that in the images produced by the method of Gormish et al. and Zhang and He there is a loss of letter details specially with the pen strokes written in green ink[3]. The image enhanced by Method A has a good contrast between background and pen strokes but some details are lost in the green pen strokes. The image enhanced in Method B has a lower contrast compared to Method A but the legibility of all the letters is preserved.

There are some cases where the loss of details can be ignored for evaluating legibility since the human visual interpolation ability can recognize partially occluded or erased characters [74]. However there are some cases where this loss of detail becomes critical to evaluate legibility. For example in Fig.37 both a phrase and the color difference ΔE^*_{ab} formula [15] are written on a whiteboard. It can be argued whether a person can recognize the content of the written phrase, but the details that are lost or poorly enhanced in the color difference formula might lead to confusion or misreadings. Once again with this dataset Method A and B offer a better legibility compared to the other 2 methods.

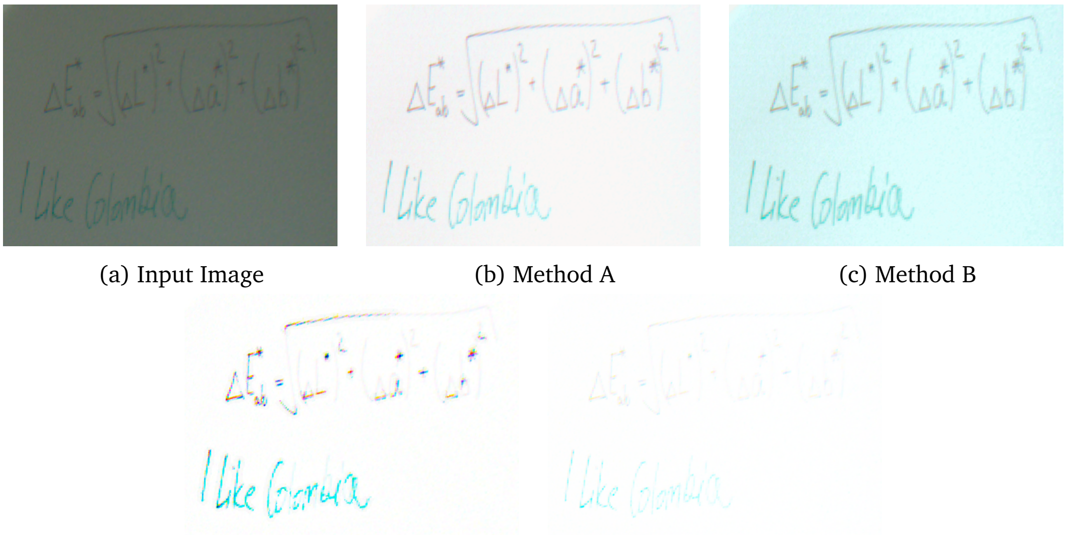
[3] We have already warned the reader about this effect in Section 4.6.



(a) Input Image (b) Method A (c) Method B

(d) Gormish et al. (e) Zhang and He

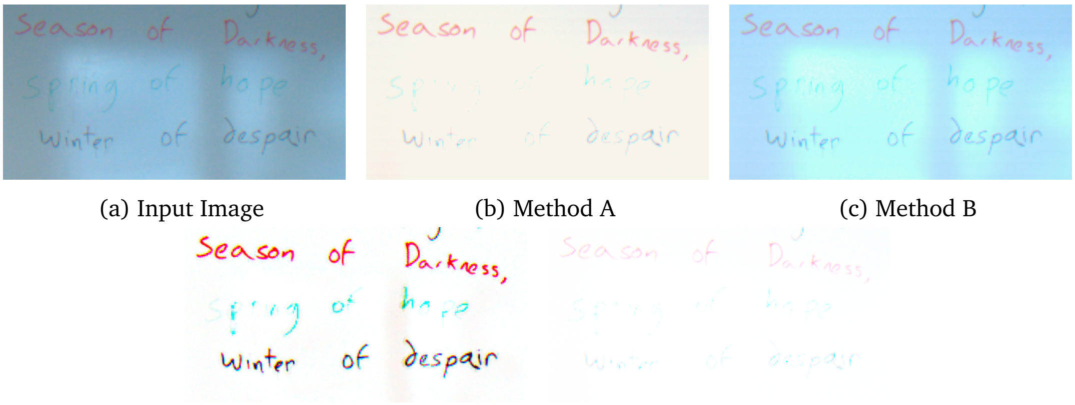
Figure 36: Legibility Results: Image set 1



(a) Input Image (b) Method A (c) Method B

(d) Gormish et al. (e) Zhang and He

Figure 37: Legibility Results: Image set 2



(a) Input Image (b) Method A (c) Method B

(d) Gormish et al. (e) Zhang and He

Figure 38: Specular Reflection: Image set 1

5.1.3 Specular Reflection

Let us see how the different algorithms deal with specular reflections on the whiteboard. As we can see in Fig.38) Method A reduces highlights on the image, however part of the pen strokes in the highlight area are missing. Since our background estimation technique ignores highlights (Section 4.21) there is a significant difference between the highlight pixels color values and their modelled background equivalents thus when we apply our contrast enhancement technique the pen-stroke color values in that area are amplified and clipped by Equation 4.21. The method of Zhang and He shows a similar behavior to Method A (This is expected since the contrast enhancement used in Method A is based on the work of Zhang and He). In Method B the highlights are slightly reduced but the pen-strokes are mostly preserved. The method of Gormish mostly ignores the specular reflection but some halos may appear since this method might segment the highlights (or areas between highlights) and process them as pen strokes.

5.1.4 Multi-illuminant

As stated in Section 3.3.1 our whiteboard analysis and enhancement implementation is based on the assumption that there is only one illuminant in the whiteboard scene or at least there is one illuminant that covers most of the whiteboard area. However in real world scenes this might not be the case. For the illumination axis estimation we assume that the whiteboard color distribution in a color space is a contained ellipsoid (Fig.10). However for an image with more of one dominant illuminant like the one in Fig40a the color distribution becomes a combination of the distribution of the different illuminants (Fig.39).

As a result our algorithm fails to compute an illumination axis that can fit this distribution. This ultimately affects the performance of Method A and Method B since method A depends on the rotation of the illumination axis (Section 4.5.1) and Method B depends on the illumination axis to calculate the color of the illuminant (Section 4.6). As a result, as we can see in Fig.40b and Fig.40c, the enhancement results become very unsatisfactory where the background is warped into different saturated colors and the pen stroke color values are changed. Since the method of

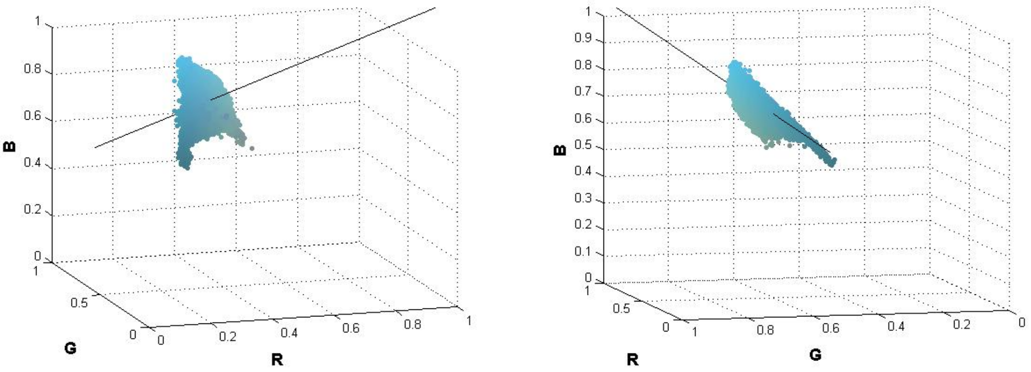


Figure 39: Whiteboard background color distribution with multiple illuminants. The black line represents the miscalculated illumination axis

Gormish et al. and the one of Zhang and He do not try to estimate the illuminants in the scene, their performances are not altered by the presence of one or more iluminants.

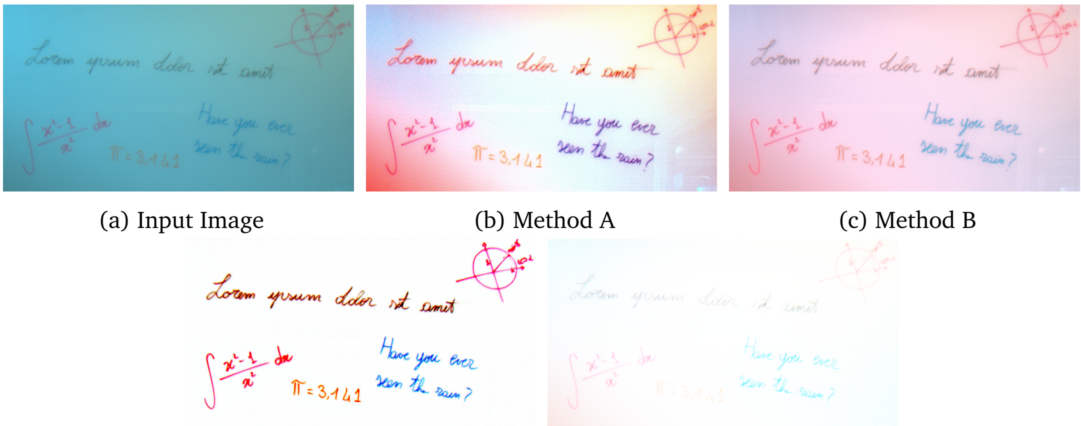
5.2 Experiment

5.2.1 Set up

We decided to evaluate the performance of our proposed system by means of psychophysical experimenting. We asked a group of human observers to grade the results of the previously discussed whiteboard enhancement techniques. The method selected was pair comparison in which an observer is given a pair of images and then he is asked to select one, according to a given criterion. The modality of this experiment is forced-choice, meaning that the observer needs to select one of the images. This method was selected due to its simplicity and since it doesn't require previous knowledge from the observer. In these experiments the observer evaluates n reproductions for m reference images and that results in $(n(n-1) \times m)/2$ comparisons. We designed two experiments:

- **Image Appearance Experiment**: In the first experiment we ask the observers which image they find more visually appealing. For this experiment we select 8 sets of test images of whiteboards which contain hand written and drawings using markers with different color inks (80 pair comparisons). After completing the experiment we asked the observers what was the criteria that they used in order to make their choices.
- **Image Legibility Experiment**: In the second experiment we ask the observers which image they find easier to read. For this experiment we select 8 sets of test images of whiteboard which contain complete hand written sentences using markers with different color inks (80 pair comparisons). After completing the experiment we asked the observers what were the characteristics they found in the more legible images.

For these experiments we created 2 whiteboard image databases. The first one is a database of whiteboard images captured in a room with controlled lighting intensity and only one type of



(a) Input Image

(b) Method A

(c) Method B

(d) Gormish et al.

(e) Zhang and He

Figure 40: Multi-illuminant case: Image set 1

illuminant. The second one has images captured in different classrooms, study rooms and offices with whiteboards all over Gjøvik University College. We created several sets of test images, each set containing the original captured image and one processed image from each enhancement method discussed in Section 5.1.

Our observers took the test in the same computer screen in the same lighting conditions using a web-based psychometric evaluation tool named QuickEval [75] provided by the Norwegian Colour and Visual Computing Laboratory.

5.2.2 Results Calculation

We got N non-expert observers for both experiments ($N = 18$ in our case). The results for each observers are stored in a $n \times n$ frequency matrix in which the value one is given in row i and column j when reproduction i is selected over reproduction j. Then a $n \times n$ summed frequency matrix can be computed by summing all raw frequency matrices. We divide the summed frequency matrix by the number of observers and we obtain a percentage matrix which shows the percentage of when a reproduction was preferred over another (e.g. table 1 and table 3). Then we obtain a logistic function matrix (LFM) by using the formula by Bartleson [75]:

$$\text{LFM} = \log\left(\frac{f + c}{N - f + c}\right) \tag{5.1}$$

where f is the value from the summed frequency matrix and c is an arbitrary additive constant ($c = 0.5$). We can transform the LFM values into z-scores using a scaling coefficient. This coefficient is calculated by taking the linear regression between the inverse of the cumulative standard normal distribution for the percentage matrix and the LFM values, where the coefficient is the slope of the regression line. The z-score matrix is found by multiplying the coefficient with the LFM. The mean z-score for each reproduction is found by taking the average of each column.

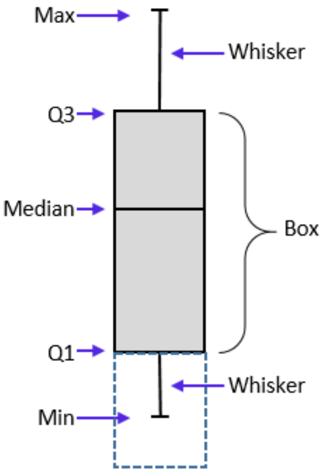


Figure 41: Simple Boxplot [77]

The z-scores are usually reported with 95% confidence intervals (CI):

$$CI = 1.96 \times \frac{\sigma}{N} \tag{5.2}$$

Since z-scores have a scale with units equal to $\sigma\sqrt{2}$, σ can be set to 1. However these results are easier to visualize using an error bar plot (e.g. Fig42a). The mean z-score value is indicated by the character in the middle of each line and the whiskers show the 95% CI. If two CIs do not overlap the difference between two mean z-scores can be considered to be significantly different with 95% confidence (and vice versa).

Another way to visualize the z-scores is with box plots (Fig.41). This is a standardized way of displaying data distribution based on five numbers: minimum, first quartile, median, third quartile and maximum. In a boxplot we draw a rectangle that spans from the first to the third quartile (The interquartile range or IQR). A segment inside the rectangle indicates the median and the whiskers above and below show the locations of the maximum and minimum [76]. The boxplot displays the full range of variation (from minimum to maximum), the likely range of variation (the IQR) and a typical value (median). The minimum is defined as 1.5 × IQR minus the first quartile and the maximum as 1.5 × IQR plus the third quartile. If a value goes below the minimum or above the maximum is considered an outlier (displayed as a cross).

5.2.3 Image Appearance Experiment Results

We can see the results of our experiment in table 1 and the computed z-scores in table 2. These results are visualized in Fig42a. The experimental results show that the reproductions made by Method A and B were preferred over all the other reproductions and their z-scores are better than the average. Although Method A and Method B got close results, Method A was preferred with 95% confidence over Method B. The method of Gormish et al. was prefered over the original reproductions but its results were worse than the average. Finally the reproductions made with the method of Zhang were the least preferred.

	Original	Method A	Method B	Gormish et al.	Zhang and He
Original	–	0.1597	0.1181	0.4653	0.8194
Method A	0.8403	–	0.7083	0.7778	0.9931
Method B	0.8819	0.2917	–	0.7222	0.9861
Gormish et al.	0.5347	0.2222	0.2778	–	0.7986
Zhang and He	0.1806	0.0069	0.0139	0.2014	–

Table 1: Image appearance percentage matrix

	Low CI limit	Mean z-score	High CI limit
Original	-0.4541	-0.3386	-0.2231
Method A	1.0767	1.1922	1.3077
Method B	0.7410	0.8565	0.9720
Gormish et al.	-0.2231	-0.1076	0.0079
Zhang and He	-1.7181	-1.6026	-1.4871

Table 2: Z-scores: Appearance experiment

As we can see in Fig.42b Method A and Method B show a similar range of variation but the IQR of Method A is in a superior range than the one in Method B (Although Method A has a narrower IQR than Method A). The method of Gormish shows a large range of variation and IQR. The original reproduction showed a small range of variation. Finally the method of Zhang shows the smallest range of variation and IQR but it has the lowest median value.

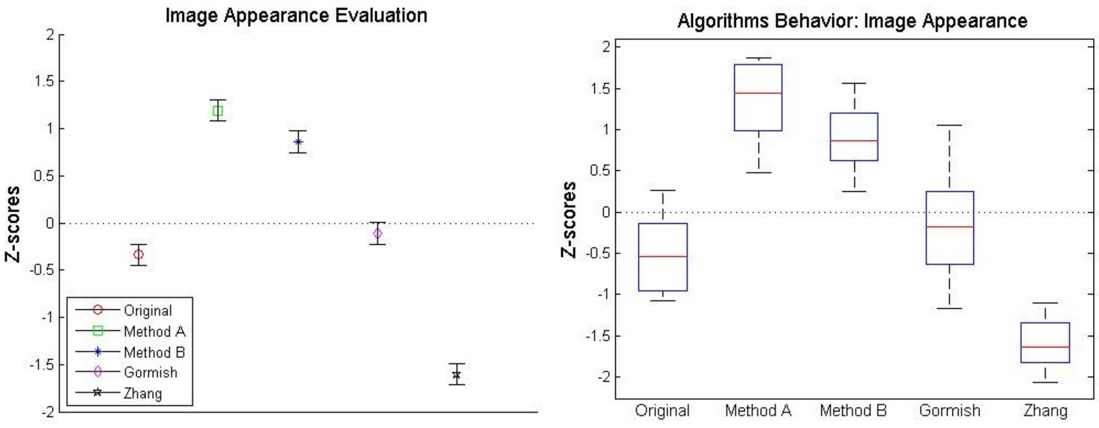
When the observers were questioned about the criteria that they used to select the image they found more visually appealing the most common answers were as follows:

- **Image Contrast**: Most of the observers agreed that the lighter the background the better the image, however they disliked images with a background that was excessively bright.
- **Legibility**: They preferred the images that were easier to read and which keep most of the pen stroke details.
- **Color Appearance**: The observers preferred images which colors were more natural. Images with artificial looking colors were disliked.

5.2.4 Legibility Experiment Results

We can see the results of our experiment in table 3 and the computed z-scores in table 4. Once again, these results are easier to visualize using the error bar plot in Fig43a. The experimental results show that the reproductions from method A and method B were preferred over any other reproduction. However this time Method B was preferred over Method A with a 95% confidence. Furthermore this time the original reproductions were preferred over the reproductions from Gormish et al. method. Finally the reproductions of Zhang and He were the least preferred.

As we can see in Fig.43b Method B has a very narrow range of distribution and IQR (with the exception of one outlier). Method A showed a big range of distribution and a medium IQR. The original reproductions showed a small and compact range of distribution and IQR(with the exception of one outlier). The method from Gormish et al. showed the largest range of distribution and IQR. Once again the method of Zhang showed a small range of distribution and



(a) Z-scores plot

(b) Z-scores boxplot

Figure 42: Z-scores: Appearance Experiment Results

	Original	Method A	Method B	Gormish et al.	Zhang and He
Original	–	0.1944	0.0417	0.6528	0.9375
Method A	0.8056	–	0.3056	0.8472	1.0000
Method B	0.9583	0.6944	–	0.9167	0.9931
Gormish et al.	0.3472	0.1528	0.0833	–	0.8264
Zhang and He	0.0625	0	0.0069	0.1736	–

Table 3: Legibility percentage matrix

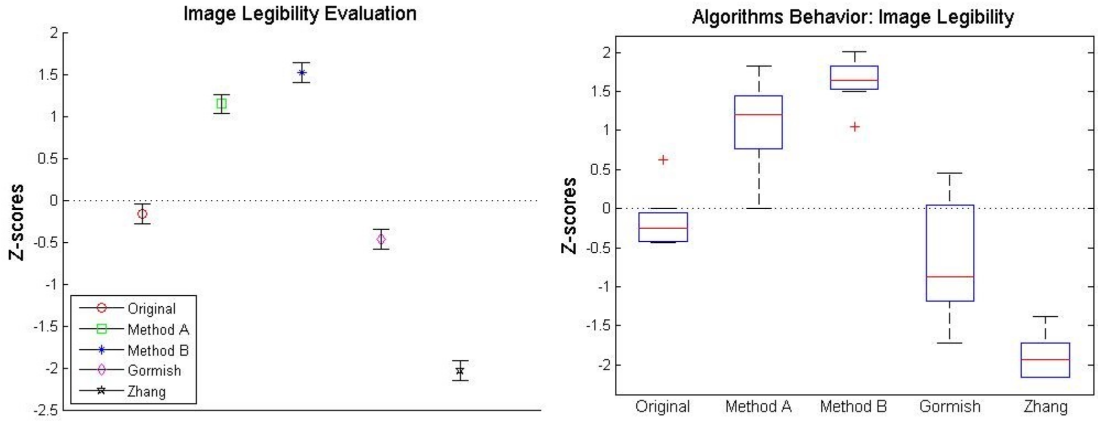
IQR.

When the observers were questioned about the criteria that they used to select which image was easier to read the most common answers were as follows:

- **Preservation of details**: The observers preferred images where most of the pen stroke details were preserved. Images with incomplete words and characters were disliked.
- **Contrast**: They stated that the lighter the background the better the image. However they preferred images with lower contrast if the pen stroke details were preserved.
- **Colour Appearance**: This was a split criteria with some observers preferring images which enhanced the pen stroke colors and others stating that color is not important for legibility.

	Low CI limit	Mean z-score	High CI limit
Original	-0.2821	-0.1666	-0.0511
Method A	1.0308	1.1463	1.2618
Method B	1.4053	1.5208	1.6363
Gormish et al.	-0.5806	-0.4651	-0.3496
Zhang and He	-2.1509	-2.0354	-1.9200

Table 4: Z-scores: Legibility experiment



(a) Z-scores plot

(b) Z-scores boxplot

Figure 43: Z-scores: Legibility Experiment Results

5.3 Data Analysis

5.3.1 Discussion on Image Appearance

The results of the image appearance experiment indicated that reproductions made with Method A and Method B were the most preferred. However Method A was consistently preferred over Method B (As evidenced by the IQRs of both methods). The reasoning behind these results might be the focus of Method A in trying to make the whiteboard neutral white by color cluster rotation resulting in an enhanced background which color values are closer to neutral white[4] rather than the darker background displayed by Method B. Furthermore, Method A seems to return pen-strokes with more saturated colors (e.g. Fig.35c compared to Fig.35d). It's worth noting that some reproductions of Method B were selected over the ones of Method A in some instances because of their legibility (As suggested by the box-plot in Fig.42b). However, as discussed in Section 5.1.4, both reproduction appearance from Method A and Method B might be affected when there are multiple illuminants in the scene.

The method of Gormish et al. showed a great variability in its results (As evidenced by its boxplot in Fig.42b). Although there might be some instances were the reproductions of Gormish et al. were preferred (e.g in Fig.32d) especially since this method gives us images with bigger contrast between pen-strokes and background, its reliability in segmentation for the enhancement process resulted in images with diminished legibility (Not to mention the "white holes" that appear within filled color shapes like the ones in Fig.33d). Furthermore, their enhancement method using an S-shaped curve(Fig.31a) resulted in images with over-saturated (and sometimes altered) colors that looked unnatural and unpleasant for the observers. These results implies that the performance of this algorithm depends on the type of content (works better with lines than color filled shapes) and scene contrast (works better when lines in the whiteboard have strong contrast w.r.t background).

[4] Although in some cases the background color might turn into a reddish tone like the one in Fig.32b.

Finally the method by Zhang was consistently the least preferred because it produced images with the lowest contrast that were more difficult to read. The resulting images suggest that their enhancement curve (Fig.31b) is able to make the whiteboard background lighter and uniform but at the cost of making the pen-strokes lighter.

5.3.2 Discussion on Image Legibility

The results on image legibility indicated that reproductions made with Method A and Method B were the most preferred. However Method B was mostly preferred over Method A (As evidenced by the IQR of both methods). Furthermore Method B shows a great consistency in their results over Method A which range of variation is relatively large. The reason for these results might be that the enhanced pen strokes with Method B are more saturated and sharper compared to the ones in Method A. Also Method B seemed to be able to preserve the details of weak pen strokes (even where there are some highlights in the scene).

Once again the method of Gormish et al. show a big variability in its results, however its reproductions were preferred even less (compared to the previous experiment)due to the loss of details in the pen-strokes. Finally the method by Zhang was the least preferred because they produced images with low contrast between pen-strokes and background making it harder to read.

The overall results of both experiments shed some light about the inherent dichotomy between image appearance and legibility when enhancing whiteboard images. Although both Method A and Method B showed to perform better than other state of the art enhancement methods it seems that we cannot conclude that one is better than the other. As indicated by our observers the criteria to decide whether an image is more visually appealing than another differs to the criteria to decide which one is more legible. Although both proposed methods partially achieved both enhancement goals it seems that depending on the whiteboard content (whether more colorful drawings or just plain characters) and the enhancement goals of a prospective user (appearance or legibility enhancement) one method should be selected over the other.

6 Conclusions

We have developed a computer vision system that captured, extracted and enhanced the content of a whiteboard applied for a videoconferencing system. This work combined different image processing and computer vision algorithms in order to create an innovative and powerful tool that better understands the color and lighting complexities of a whiteboard.

Thanks to an extensive review of the state of the art in whiteboard image capture and enhancement systems and a deep analysis of the variations of color behavior and distribution along a whiteboard surface and color spaces we were able to design a series of methods for measuring, modelling and finally enhancing the whiteboard content.

We have developed a robust background estimation technique that is able to model complex lighting variations in presence of specular reflection. This eventually allowed us to lighten the whiteboard background and identify occlusions.

We were able to model the whiteboard color distribution with the illumination axis. This allowed us not only to identify the color values of the illuminant and the pen strokes but also became the basis for our enhancement methodology.

We have proposed two different enhancement methods. The first one focused on enhancing image appearance by making the background lighter by means of color cluster rotation. The second method focused on preserving image legibility by making the background closer to the color of the estimated illuminant.

We were able to experimentally show the complex dichotomy between image appearance and image legibility goals and we were able to report on the criteria and challenges in order to develop a method in the future that would be able to achieve a satisfactory middle ground between both objectives.

6.1 Future Work

Although we were able to develop a functional and robust whiteboard content extraction and enhancement system there are still many opportunities to extend the scope of this thesis. Here are some of the areas that could be explored for future work:

- **Image enhancement Method C**: Explore different combinations between our proposed enhancement methods A and B and implement a method that provide images with whiter background and more saturated pen strokes while preserving the legibility of its content.
- **Automatic whiteboard detection**: So far in our system the user has to click on the corners of the whiteboard in order to isolate the whiteboard area. These corners could be easily identifiable using the Harris corner detection method [78]. Furthermore, since a whiteboard can be seen as a quadrangle, we could detect the whiteboard area by means of edge detection and using the Hough transform to detect straight lines [22].
- **Multi-illuminant color estimation**: In order to improve the performances of our enhance-

ment methods we could estimate the colors of multiple illuminants and their spatial distribution in the scene. There are some methods that segment the image into patches and estimate the illuminant patchwise [79]. Other methods formulate this problem as an energy minimization task within a conditional random field over a set of local illuminant estimates [80].

- **Improve image cell classification**: The first improvement would be to increase the classification speed using parallel programming. Each image cell can be mapped to a thread, making it possible to process them simultaneously. The second improvement would be to represent the cells as nodes in a graph in order to identify adjacent cells as neighbors and apply neighborhood operations on the cells.
- **Specular Reflection Removal**: Since our background estimation method ignores highlights, the image-cell classification could be modified in order to identify those areas. Then we could review and apply different methods in order to separate and remove the specular reflection component of our whiteboard images [81].
- **Pen Stroke Color Enhancement**: We could review different hue preserving color image enhancement techniques [82][83] and compare their performance with the color warping algorithm used for pen stroke color enhancement.

Bibliography

[1] Commitment AB. Types of communication medium. http://www.communicationtoolbox.com/types_of_communication_medium.html. Accesed: 2015-05-28.

[2] Beal, V. videoconferencing. http://www.webopedia.com/TERM/V/videoconferencing.html. Accesed: 2015-05-28.

[3] kubicam. kubicam. http://kubicam.com/. Accesed: 2015-01-30.

[4] Lu, H. & Kowalkiewicz, M. Dec 2012. Text segmentation in unconstrained hand-drawings in whiteboard photos. In *Digital Image Computing Techniques and Applications (DICTA), 2012 International Conference on*, 1–6. doi:10.1109/DICTA.2012.6411687.

[5] Plötz, T., Thurau, C., & Fink, G. A. 2008. Camera-based whiteboard reading: New approaches to a challenging task. In *Proc. Int. Conf. on Frontiers in Handwriting Recognition*, 385–390, Montreal, Canada.

[6] Wienecke, M., Fink, G., & Sagerer, G. Aug 2003. Towards automatic video-based whiteboard reading. In *Document Analysis and Recognition, 2003. Proceedings. Seventh International Conference on*, 87–91 vol.1. doi:10.1109/ICDAR.2003.1227633.

[7] Zhang, Z. & Tan, C. 2001. Restoration of images scanned from thick bound documents. In *Image Processing, 2001. Proceedings. 2001 International Conference on*, volume 1, 1074–1077 vol.1. doi:10.1109/ICIP.2001.959235.

[8] Vajda, S., Rothacker, L., & Fink, G. 2012. A method for camera-based interactive whiteboard reading. In *Camera-Based Document Analysis and Recognition*, Iwamura, M. & Shafait, F., eds, volume 7139 of *Lecture Notes in Computer Science*, 112–125. Springer Berlin Heidelberg. URL: http://dx.doi.org/10.1007/978-3-642-29364-1_9, doi:10.1007/978-3-642-29364-1_9.

[9] Fakih, A. & Serrao, J. 2012. Real-time whiteboard capture system. *IOSR Journal of Electrical and Electronics Engineering*, 2(1), 21–25.

[10] Chandika Dana Siva Sankara Rao, Kanuri Venkata Chaitanya, K. S. K. & Ramesh, G. 2013. Whiteboard image reconstruction using matlab. *International Journal of Research in Engineering and Technology*, 3(07), 255–260.

[11] Dickson, P., Adrion, W., & Hanson, A. Dec 2008. Whiteboard content extraction and analysis for the classroom environment. In *Multimedia, 2008. ISM 2008. Tenth IEEE International Symposium on*, 702–707. doi:10.1109/ISM.2008.35.

[12] Xu, R. Jan 2008. A computer vision based whiteboard capture system. In *Applications of Computer Vision, 2008. WACV 2008. IEEE Workshop on*, 1–6. doi:10.1109/WACV.2008.4544028.

[13] Sakshuwong, S. & Tsai, B. 2014. Whiteboard image extraction and archival tool implementation in iphone application. http://web.stanford.edu/class/ee368/Project_Autumn_1314/. Department of Electrical Engineering, Stanford University.

[14] Otsu, N. Jan 1979. A threshold selection method from gray-level histograms. *Systems, Man and Cybernetics, IEEE Transactions on*, 9(1), 62–66. doi:10.1109/TSMC.1979.4310076.

[15] Ohta, N. & Robertson, A. 2005. *Colorimetry: Fundamentals and Applications*. Wiley.

[16] He, Y., Sun, J., Naoi, S., Minagawa, A., & Hotta, Y. 2010. Enhancement of camera-based whiteboard images. volume 7534 of *SPIE Proceedings*, 1–10. SPIE. URL: http://dblp.uni-trier.de/db/conf/drr/drr2010.html#HeSNMH10.

[17] Shafer, S. A. 1985. Using color to separate reflection components. *Color Research and Application*, 10(4), 210–218. URL: http://dx.doi.org/10.1002/col.5080100409, doi:10.1002/col.5080100409.

[18] Tan, R. & Ikeuchi, K. Feb 2005. Separating reflection components of textured surfaces using a single image. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(2), 178–193. doi:10.1109/TPAMI.2005.36.

[19] Gormish, M., Erol, B., Van Olst, D. G., Li, T., & Mariotti, A. 2011. Whiteboard sharing: capture, process, and print or email. volume 7879, 78790D–78790D–9. URL: http://dx.doi.org/10.1117/12.872162, doi:10.1117/12.872162.

[20] Canny, J. Nov 1986. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-8(6), 679–698. doi:10.1109/TPAMI.1986.4767851.

[21] Zhang, Z. & He, L. W. March 2007. Whiteboard scanning and image enhancement. *Digit. Signal Process.*, 17(2), 414–432. URL: http://dx.doi.org/10.1016/j.dsp.2006.05.006, doi:10.1016/j.dsp.2006.05.006.

[22] Zhang, Z. & He, L. W. Notetaking with a camera: Whiteboard scanning and image enhancement. Technical report, Microsoft Research, 2003.

[23] He, L. W., Liu, Z., & Zhang, Z. Why take notes? use the whiteboard capture system. Technical Report MSR-TR-2002-89, Microsoft Research, September 2002. URL: http://research.microsoft.com/apps/pubs/default.aspx?id=69966.

[24] Boozer, S. Pixid whiteboard photo. http://www.imaging-resource.com/SOFT/WBS/WBSA.HTM. Accesed: 2015-05-28.

[25] Keller, J. Whiteboard photo review. http://www.dcresource.com/specials/WhiteBoardPhoto/. Accesed: 2015-05-28.

[26] SoftTouch IT. Why take notes if you can clearboard? http://www.softtouchit.com/products/clearboard. Accesed: 2015-05-28.

[27] snapclean.me. http://snapclean.me/. Accesed: 2015-05-28.

[28] (un)whiteboard. http://www.unwhiteboard.com/. Accesed: 2015-05-28.

[29] Chuang, D. How to clean up whiteboard photos with apps. https://djchuang.wordpress.com/2013/05/22/how-to-clean-up-whiteboard-photos-with-apps/. Accesed: 2015-05-28.

[30] The Center of Graphic Facilitation. November 2006. Cleaning digital photos of drawings. http://graphicfacilitation.blogs.com/pages/2006/11/cleaning_digita.html. Accesed: 2015-05-28.

[31] wikiHow. How to copy a whiteboard with your digital camera or camera phone. http://www.wikihow.com/Copy-a-Whiteboard-With-Your-Digital-Camera-or-Camera-Phone. Accesed: 2015-05-28.

[32] Golovchinsky, G., Carter, S., & Biehl, J. 2009. Beyond the drawing board: Toward more effective use of whiteboard content". arxiv preprint arxiv:0911.0039.

[33] Branham, S., Golovchinsky, G., Carter, S., & Biehl, J. T. 2010. Let's go from the whiteboard: Supporting transitions in work through whiteboard capture and reuse. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, 75–84, New York, NY, USA. ACM. URL: http://doi.acm.org/10.1145/1753326.1753338, doi:10.1145/1753326.1753338.

[34] BBC Active. What is an interactive whiteboard? http://www.bbcactive.com/BBCActiveIdeasandResources/Whatisaninteractivewhiteboard.aspx. Accesed: 2015-05-28.

[35] Soares, C., Moreira, R. S., Torres, J. M., & Sobral, P. 2013. Locoboard: Low-cost interactive whiteboard using computer vision algorithms. *ISRN Machine Vision*, 2013.

[36] SMART. Hardware. http://education.smarttech.com/en/products/hardware. Accesed: 2015-05-28.

[37] Promethean. Activboard touch. http://www.prometheanworld.com/us/english/education/products/interactive-whiteboard-systems/activboard-touch/. Accesed: 2015-05-28.

[38] Luidia Inc. Activboard touch. http://www.e-beam.com/home.html. Accesed: 2015-05-28.

[39] Panasonic. What's panaboard. http://panasonic.net/pcc/eboard/whats_panaboard/index.html. Accesed: 2015-05-28.

[40] Camplani, M., Salgado, L., & Camplani, R. Jan 2012. Low-cost efficient interactive whiteboard. In *Consumer Electronics (ICCE), 2012 IEEE International Conference on*, 686–687. doi:10.1109/ICCE.2012.6161809.

[41] Microsoft. Microsoft surface hub. https://www.microsoft.com/microsoft-surface-hub/en-us. Accesed: 2015-05-28.

[42] Spilka, S. Microsoft reinvents the traditional whiteboard. http://www.psfk.com/2015/01/microsoft-reinvents-traditional-whiteboard.html. Accesed: 2015-05-28.

[43] McClean, G. Whiteboards' history, materials and buying tips. http://www.theworkplacedepot.co.uk/news/2014/08/20/whiteboards-history/. Accesed: 2015-03-26.

[44] Larramendi, C. H., Marco, F. M., Llombart, M., de la Vega, A., Chiner, E., García-Abujeta, J. L., & Sempere, J. M. 2013. Allergenicity of casein containing chalk in milk allergic schoolchildren. *Annals of Allergy, Asthma & Immunology*, 110(5), 335 – 339. URL: http://www.sciencedirect.com/science/article/pii/S1081120613001002, doi:http://dx.doi.org/10.1016/j.anai.2013.02.006.

[45] Ramani, J. Advantages and disadvantages of a whiteboard. http://ezinearticles.com/?Advantages-and-Disadvantages-of-a-Whiteboard&id=6386214. Accesed: 2015-03-26.

[46] Arana, W. How are dry erase markers made? http://www.ehow.com/how-does_4965235_how-dry-erase-markers-made.html. Accesed: 2015-03-26.

[47] wiseGEEK. What are dry erase markers? http://www.ehow.com/how-does_4965235_how-dry-erase-markers-made.html. Accesed: 2015-03-26.

[48] Brown, E. The disadvantages of whiteboards in the classroom. http://www.ehow.com/list_5805781_disadvantages-whiteboards-classroom.html. Accesed: 2015-06-28.

[49] Brainard DH, Brunt WA, S. J. 1997. Color constancy in the nearly natural image. i. asymmetric matches. *Journal of Optical Society of America*, 14(9), 2091–2110.

[50] Sayers, N. Tips for filming whiteboard presentations. https://moz.com/blog/tips-for-filming-whiteboard-presentations. Accesed: 2015-03-26.

[51] Tomazevic, D., Likar, B., & Pernus, F. 2000. A comparison of retrospective shading correction techniques. In *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, volume 3, 564–567 vol.3. doi:10.1109/ICPR.2000.903608.

[52] Münzenmayer, C., Naujokat, F., Mühldorfer, S., Erlangen-nürnberg, U., I, M. K., & D-Erlangen. 2003. Enhancing texture analysis by color shading correction. In *in 9. Workshop Farbbildverarbeitung*.

[53] Russ, J. C. 2002. *Image Processing Handbook, Fourth Edition*. CRC Press, Inc., Boca Raton, FL, USA, 4th edition.

[54] Rousseeuw, P. J. 1984. Least Median of Squares Regression. *Journal of The American Statistical Association*, 79, 871–880. doi:10.1080/01621459.1984.10477105.

[55] Zhang, Z. Least Median of Squares. http://research.microsoft.com/en-us/um/people/zhang/inria/publis/tutorial-estim/node25.html#SECTION000105000000000000000. Accesed: 2015-03-27.

[56] Rousseeuw, P. J. & Leroy, A. M. 1987. *Robust Regression and Outlier Detection*. John Wiley & Sons, Inc., New York, NY, USA.

[57] Paulus, D., Csink, L., & Niemann, H. Oct 1998. Color cluster rotation. In *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*, volume 1, 161–165 vol.1. doi:10.1109/ICIP.1998.723449.

[58] Park, J. B. 2003. Efficient color representation for image segmentation under nonwhite illumination. In *In SPIE, Volume 5267*, 163–174.

[59] Park, J. B. *A New Color Representation for Non-white Illumination Conditions: An Effective Approach to Color Machine Vision*. PhD thesis, West Lafayette, IN, USA, 2004. AAI3166691.

[60] Wolfram MathWorld. Projection. http://mathworld.wolfram.com/Projection.html. Accesed: 2015-06-06.

[61] Point Grey. Flea3 specifications. http://www.ptgrey.com/support/downloads/10136. Accesed: 2015-06-06.

[62] Point Grey. Understanding format_7 region of interest and pixel binning modes. https://www.ptgrey.com/KB/10085. Accesed: 2015-06-06.

[63] Point Grey. Flea3. http://www.ptgrey.com/flea3-usb3-vision-cameras. Accesed: 2015-06-06.

[64] Brafford, K. pyfly2. https://github.com/kbrafford/pyfly2. Accesed: 2015-01-30.

[65] endolith. peakdet.py. http://kubicam.com/. Accesed: 2015-01-30.

[66] Wren, C. R. Perspective transform estimation. http://xenia.media.mit.edu/~cwren/interpolator/. Accesed: 2015-03-27.

[67] Criminisi, A., Reid, I., & Zisserman, A. 1999. A plane measuring device. *Image and Vision Computing*, 17(8), 625 – 634. URL: http://www.sciencedirect.com/science/article/pii/S0262885698001838, doi:http://dx.doi.org/10.1016/S0262-8856(98)00183-8.

[68] Getreuer, P. 2011. Linear Methods for Image Interpolation. *Image Processing On Line*, 1. doi:10.5201/ipol.2011.g_lmii.

[69] Tomasi, C. & Manduchi, R. Jan 1998. Bilateral filtering for gray and color images. In *Computer Vision, 1998. Sixth International Conference on*, 839–846. doi:10.1109/ICCV.1998.710815.

[70] Perona, P. & Malik, J. July 1990. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(7), 629–639. URL: http://dx.doi.org/10.1109/34.56205, doi:10.1109/34.56205.

[71] Faugeras, O. 1993. *Three-dimensional Computer Vision: A Geometric Viewpoint*. MIT Press, Cambridge, MA, USA.

[72] Hardeberg, J. Y., Farup, I., Kolåss, Ø., & Stjernvang, G. Sep 2002. Color management for digital video: Color correction in the editing phase. In *29th International iarigai Research Conference. Proceedings: Advances in Graphic Arts & Media Technology*, Lucerne, Switzerland.

[73] Avraam, M. Readability vs. legibility. http://michalisavraam.org/2009/05/readability-vs-legibility/. Accesed: 2015-06-28.

[74] Jiang, Y. & Wang, S. Sept 2007. Measurement and quantitative analysis of the human visual interpolation ability for partially erased objects. In *Innovative Computing, Information and Control, 2007. ICICIC '07. Second International Conference on*, 4–4. doi:10.1109/ICICIC.2007.378.

[75] Ngo, K. V., Storvik, J. J., Dokkeberg, C. A., Farup, I., & Pedersen, M. Feb 2015. QuickEval: A web application for psychometric scaling experiments. In *Image Quality and System Performance XII*, volume 9396 of *Proceedings of SPIE/IS&T Electronic Imaging*, 9396–24, San Francisco, CA, USA. SPIE. URL: http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=2109944.

[76] Kirkman, T. W. Box plot: Display of distribution. http://www.physics.csbsju.edu/stats/box2.html. Accesed: 2015-06-28.

[77] Dalgleish, D. Create a simple box plot in excel. http://blog.contextures.com/archives/2013/06/11/create-a-simple-box-plot-in-excel/. Accesed: 2015-06-28.

[78] Harris, C. & Stephens, M. 1988. A combined corner and edge detector. In *In Proc. of Fourth Alvey Vision Conference*, 147–151.

[79] Gijsenij, A., Lu, R., & Gevers, T. Feb 2012. Color constancy for multiple light sources. *Image Processing, IEEE Transactions on*, 21(2), 697–707. doi:10.1109/TIP.2011.2165219.

[80] Beigpour, S., Riess, C., van de Weijer, J., & Angelopoulou, E. Jan 2014. Multi-illuminant estimation with conditional random fields. *Image Processing, IEEE Transactions on*, 23(1), 83–96. doi:10.1109/TIP.2013.2286327.

[81] Artusi, A., Banterle, F., & Chetverikov, D. 2011. A survey of specularity removal methods. *Computer Graphics Forum*, 30(8), 2208–2230. URL: http://dx.doi.org/10.1111/j.1467-8659.2011.01971.x, doi:10.1111/j.1467-8659.2011.01971.x.

[82] Naik, S. & Murthy, C. Dec 2003. Hue-preserving color image enhancement without gamut problem. *Image Processing, IEEE Transactions on*, 12(12), 1591–1598. doi:10.1109/TIP.2003.819231.

[83] Gorai, A. & Ghosh, A. Sept 2011. Hue-preserving color image enhancement using particle swarm optimization. In *Recent Advances in Intelligent Computational Systems (RAICS), 2011 IEEE*, 563–568. doi:10.1109/RAICS.2011.6069375.

A Experiment Instructions

Whiteboard Content Enhancement Experiments

Motivation: We are developing a method to extract and enhance the content of a whiteboard captured by a videocamara. In order to test its efficacy we are going to test it against other state of the art algorithms and let you decide which one is better.

Methodology: The method we are going to use is called pair comparison. It's a very simple procedure where an observer looks at two pictures and selects one of them according to certain criteria.

A.1 Instructions

A.1.1 First Experiment

1. Go to QuickEval website: http://www.ansatt.hig.no/mariusp/quick/index.php
2. Log in as Anonymous or as member of the colourlab if you already have an account.
3. Click the box "Scientist" below the title "Select Experiment". Select Scientist "Arango, Carlos".
4. Click in the experiment "Whiteboard Content Enhancement - Appearance Test". In the bottom part of the webpage please fill the input field "Age" and click on the button "Start Experiment".
5. Read the tutorial instruction.
6. You will see two different images like this:

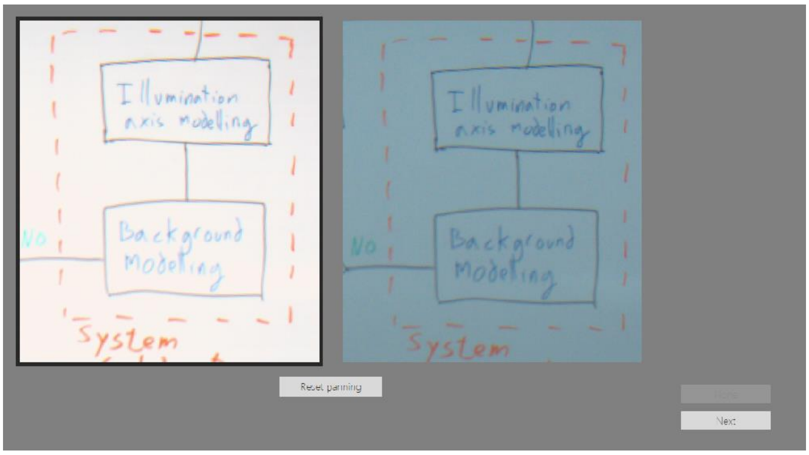


Figure 44: Instruction of QuickEval

7. Select the image that you prefer.

8. Click on the button “Next”.
9. Repeat the 2 previous steps until you finish the tutorial
10. Let’s start the experiment. You will have a group of image pairs. Select the image that you find more visually appealing.
11. Continue until you finish the experiment.
12. Take a 5-10 minutes break

A.1.2 Second Experiment

1. Repeat steps 1 to 3 from the previous experiment.
2. Click in the experiment “Whiteboard Content Enhancement - Legibility Test”. In the bottom part of the webpage please fill the input field “Age” and click on the button “Start Experiment”.
3. You will have a group of image pairs. Select the image that you find easier to read.
4. Continue until you finish the experiment

A.1.3 Survey

Please answer the following questions:

- During the first experiment why do you select one image over the other? What were the characteristics in the image that you considered more visually appealing?
- During the second experiment why do you consider that one image was easier to read than the other? What were the aspects that you considered to make a decision?

Thank you very much for participating in this experiment!